

LNBIP 20

Jianhua Yang
Athula Ginige
Heinrich C. Mayr
Ralf-D. Kutsche (Eds.)

Information Systems: Modeling, Development, and Integration

Third International United Information Systems Conference
UNISCON 2009
Sydney, Australia, April 2009, Proceedings

Lecture Notes in Business Information Processing

20

Series Editors

Wil van der Aalst

Eindhoven Technical University, The Netherlands

John Mylopoulos

University of Trento, Italy

Norman M. Sadeh

Carnegie Mellon University, Pittsburgh, PA, USA

Michael J. Shaw

University of Illinois, Urbana-Champaign, IL, USA

Clemens Szyperski

Microsoft Research, Redmond, WA, USA

Jianhua Yang Athula Ginige
Heinrich C. Mayr Ralf-D. Kutsche (Eds.)

Information Systems: Modeling, Development, and Integration

Third International United Information Systems Conference
UNISCON 2009
Sydney, Australia, April 21-24, 2009
Proceedings

Volume Editors

Jianhua Yang
Athula Ginige
University of Western Sydney
Parramatta, NSW 2150, Australia
E-mail: {j.yang,a.ginige}@uws.edu.au

Heinrich C. Mayr
University of Klagenfurt
9020 Klagenfurt, Austria
E-mail: heinrich.mayr@uni-klu.ac.at

Ralf-D. Kutsche
Berlin University of Technology
10587 Berlin, Germany
E-mail: rkutsche@cs.tu-berlin.de

Library of Congress Control Number: Applied for

ACM Computing Classification (1998): H.3, H.4, J.1, D.2

ISSN 1865-1348
ISBN-10 3-642-01111-X Springer Berlin Heidelberg New York
ISBN-13 978-3-642-01111-5 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

springer.com

© Springer-Verlag Berlin Heidelberg 2009
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12643790 06/3180 5 4 3 2 1 0

Preface

UNISCON 2009 (United Information Systems Conference) was the third conference in the series that is based on the idea to pool smaller but highly interesting scientific events on information systems into one large conference. Here, people from different scientific backgrounds can present their research results, share their ideas and discuss future trends in these various areas. UNISCON 2009 was held in Sydney, Australia in the University of Western Sydney, Campbelltown Campus.

In 2009 the following scientific events were held under the umbrella of UNISCON 2009:

- 8th International Conference on Information Systems Technology and Its Applications (ISTA 2009)
- 8th International Workshop on Conceptual Modelling Approaches for e-Business (eCOMO 2009)
- Second Workshop on Model-Based Software and Data Integration (MBSDI 2009)

We received 115 papers for the three events. Papers were submitted from over 25 countries. After a rigorous review process, 39 papers were accepted as full papers and 14 papers as short papers for presentation at the conference and published in these proceedings.

In addition to the above three events, we also organized a Doctoral Consortium to provide a forum for doctoral students to get feedback from experts in the area about their research projects.

Many people worked very hard to make UNISCON 2009 a success. Among them we would especially like to thank Anupama Ginige, Christian Kop, Jianhua (Jamie) Yang, Simi Bajaj and Ana Hol for all their detailed planning and organizing of various aspects of this conference. We would also like to thank the ISTA 2009 PC Co-chairs Christian Kop, Simeon Simoff and Stephen W. Liddle, eCOMO 2009 PC Co-chairs San Murugesan, Christian Kop and Dimitris Karagiannis, MBSDI 2009 PC Co-chairs Ralf-D. Kutsche and Nikola Milanovic as well as the Program Committees of the three events for their contributions. Their names and affiliations are listed separately. We would also like to thank the other members of the local Organizing Committee for their various contributions toward making UNISCON 2009 a success. We would also like to express our gratitude to the keynote speakers, invited speakers and members of the panels for accepting our invitation to participate in the conference and for sharing their knowledge with the other participants. In addition, we would also like to show our appreciation to all the authors who submitted papers for the three events and the authors of accepted papers who came to the conference to discuss and share their findings with others.

Finally, we like to thank all the UNISCON 2009 attendees and all those who contributed in various ways to making this conference a success.

April 2009

Athula Ginige
Heinrich C. Mayr
Ralf-D. Kutsche

Organization

Organization Committee

General Chairs	Athula Ginige, University of Western Sydney, Australia Heinrich C. Mayr, University of Klagenfurt, Austria Ralf-D. Kutsche, TU Berlin, Germany
ISTA Co-chairs	Christian Kop, University of Klagenfurt, Austria Simeon Simoff, University of Western Sydney, Australia Stephen W. Liddle, Rollins Centre for eBusiness, USA
eCOMO Co-chairs	San Murugesan, Multimedia University, Malaysia Christian Kop, University of Klagenfurt, Austria Dimitris Karagiannis, University of Vienna, Austria
MBSDI Co-chairs	Ralf-D. Kutsche, TU Berlin, Germany Nikola Milanovic, TU Berlin, Germany
Local Organizing Chair	Anupama Ginige, University of Western Sydney,Australia
Publicity Co-chairs	Ana Hol, University of Western Sydney, Australia Vladimir Shekhovtsov, National Technical University (KhPI), Kharkiv, Ukraine
Finance and Sponsorship Chair	Simi Bajaj, University of Western Sydney, Australia
Proceedings Publication Chair	Jianhua (Jamie) Yang, University of Western Sydney, Australia
Industry Panel Chair	Olivera Marjanovic, University of Sydney

ISTA Program Committee

Witold Abramowicz	The Poznan University of Economics, Poland
Marko Anzelak	Alpen-Adria-Universität Klagenfurt, Austria
Stephane Bressan	National University of Singapore, Singapore
Ruth Breu	University of Innsbruck, Austria
Anatoly Doroshenko	Institute of Software Systems, Ukraine
Vadim Ermolayev	Zaporozhye State University, Ukraine
Günther Fliedl	Alpen-Adria-Universität Klagenfurt, Austria
Alexander Gelbukh	National Polytechnic Institute, Mexico
Robert Goodwin	Flinders University, Adelaide, Australia
Remigijus Gustas	Karlstad University, Sweden
Jean Luc Hainaut	University of Namur, Belgium
Wolfgang Hesse	Philipps University Marburg, Germany
Wolfgang H. Janko	University of Economics and Business Administration, Austria
Roland Kaschek	KIMEP in Almaty, Kazakhstan
Jörg Kerschbaumer	Alpen-Adria-Universität Klagenfurt, Austria
Thomas Kühne	Victoria University Wellington, New Zealand
Myoung Ho Kim	KAIST, Korea
Kinshuk	Athabasca University, Canada
Mikhail R. Kogalovsky	Market Economy Institute RAS, Russia
Elisabeth Metais	Laboratory CEDRIC, Paris, France
San Murugesan	Multimedia University, Malaysia
Igor Nekrestyanov	St. Petersburg State University, Russia
Boris A. Novikov	St. Petersburg State University, Russia
Andreas Oberweis	AIFB Karlsruhe (TH), Germany
Maria E. Orlowska	The University of Queensland, Australia
Wolfgang Reisig	Humboldt-Universität zu Berlin, Germany
Bernhard Rumpe	TU Braunschweig, Germany
Gunter Saake	Otto von Guericke University, Magdeburg, Germany
Joachim Schmidt	TU Hamburg Harburg, Germany
Vladimir Shekhovtsov	National Technical University “KhPI”, Kharkiv, Ukraine
Claudia Steinberger	Alpen-Adria-Universität Klagenfurt, Austria
Keng Siau	University of Nebraska-Lincoln, USA
Markus Stumptner	University of South Australia, Australia
Yuzura Tanaka	Hokkaido University, Japan
Helmut Thoma	Universität Basel, Switzerland
Jürgen Vöhringer	Alpen-Adria-Universität Klagenfurt, Austria
Vyacheslav Wolfengagen	JurInfoR-MSU, Russia
Joze Zupancic	University of Maribor, Slovenia
Graeme Salter	University of Western Sydney, Australia
Filomena Ferrucci	University of Salerno, Italy
Peter Bellström	Karlstad University, Sweden

eCOMO Program Committee

Aditya Ghose	University of Woolongong, Australia
Angappan Gunasekaran	University of Massachusetts, USA
Athula Ginige	University of Western Sydney, Australia
Bernhard Thalheim	Christian Albrechts University Kiel, Germany
Bhuvan Unkelhar	MethodsScience.Com, Australia
Bill Karakostas	UMIST Manchester, UK
Christian Kop	Alpen-Adria-Universität Klagenfurt, Austria
Chun Ruan	University of Western Sydney, Australia
Doris Gälle	Alpen-Adria-Universität Klagenfurt, Austria
Elmar Sinz	Universität Bamberg, Germany
Fahim Akhter	Zayed University, United Arab Emirates
Farouk Toumani	Blaise Pascale University, France
Gerti Kappel	Technical University of Vienna, Austria
Giuliana Vitiello	University of Salerno, Italy
Hui Ma	University of Wellington, New Zealand
Il-Yeol Song	Drexel University, Philadelphia, USA
Jan Jürjens	The Open University, UK
Jian Yang	Macquarie University Sydney, Australia
Joan Fons	Valencia University of Technology, Spain
Jon Atle Gulla	Norwegian University of Science and Technology, Norway
Jörg Desel	Katholische Universität, Eichstätt, Germany
Jos van Hillegersberg	Erasmus University, The Netherlands
Marcela Genero	University of Castilla-La Mancha, Spain
Matti Rossi	Helsinki School of Economics, Finland
Oscar Pastor	Valencia University of Technology, Spain
Reind van de Riet	Vrije Universiteit Amsterdam, The Netherlands
Roland Kaschek	KIMEP, Almaty, Kazakhstan
Sudha Ram	University of Arizona, USA
Tatjana Welzer	University of Maribor, Slovenia
Vadim A. Ermolayev	Zaporozhye State University, Ukraine
Vijay Sugumaran	Oakland University, USA
Willem Jan-van den Heuvel	Tilburg University, The Netherlands
Yogesh Deshpande	University of Western Sydney, Australia

MBSDI Program Committee

Andreas Billig	Jönköping University, Sweden
Susanne Busse	TU Berlin, Germany
Tru Hoang Cao	HCMUT, Vietnam
Stefan Conrad	University of Düsseldorf, Germany
Bich-Thuy T. Dong	HCMUNS, Vietnam
Anupama Ginige	University of Western Sydney, Australia
Michael Goedicke	University of Duisburg-Essen, Germany

Martin Große-Rhode	Fraunhofer ISST, Berlin, Germany
Oliver Günther	HU Berlin, Germany
Willi Hasselbring	University of Kiel, Germany
Maritta Heisel	University of Duisburg-Essen, Germany
Arno Jacobsen	University of Toronto, Canada
Ralf-D. Kutsche	TU Berlin, Germany
Andreas Leicher	Carneq GmbH, Berlin, Germany
Michael Löwe	FHDW, Hannover, Germany
Nikola Milanovic	TU Berlin, Germany
Andreas Polze	HPI, Potsdam, Germany
Ralf Reussner	University of Karlsruhe, Germany
Premaratne Samaranayake	University of Western Sydney, Australia
Kurt Sandkuhl	Jönköping University, Sweden
Alexander Smirnov	SPIIRAS, St. Petersburg, Russia
Jun Suzuki	University of Massachusetts, Boston
Stefan Tai	IBM Yorktown Heights, USA
Bernhard Thalheim	University of Kiel, Germany
Daniel Varro	University of Budapest, Hungary
Gregor Wolf	Klopotek AG, Berlin, Germany
Katinka Wolter	HU Berlin, Germany
Uwe Zdun	TU Wien, Austria
Joe Zou	IBM, Australia

Table of Contents

Keynotes (Abstracts)

Model-Based Design - A Chance for Good Quality Products and Services by Integrating Intelligence	1
<i>András Pataricza</i>	
Process Modelling - What Really Matters	3
<i>Michael Rosemann</i>	

ISTA Papers

Information Systems Modelling

Robust Web Services Provisioning through On-Demand Replication	4
<i>Quan Z. Sheng, Zakaria Maamar, Jian Yu, and Anne H.H. Ngu</i>	
Modelling and Maintenance of Very Large Database Schemata Using Meta-structures	17
<i>Hui Ma, Klaus-Dieter Schewe, and Bernhard Thalheim</i>	
SOM-Based Dynamic Image Segmentation for Sign Language Training Simulator	29
<i>Oles Hodych, Kostiantyn Hushchyn, Yuri Shcherbyna, Iouri Nikolski, and Volodymyr Pasichnyk</i>	
Agile Software Solution Framework: An Analysis of Practitioners' Perspectives	41
<i>Asif Qumer and Brian Henderson-Sellers</i>	
Facilitating Inter-organisational Collaboration via Flexible Sharing of Rapidly Developed Web Applications	53
<i>Ioakim (Makis) Marmaridis, Xufeng (Danny) Liang, and Athula Ginige</i>	
Sales Forecasting Using an Evolutionary Algorithm Based Radial Basis Function Neural Network	65
<i>R.J. Kuo, Tung-Lai Hu, and Zhen-Yao Chen</i>	
Micro Implementation of Join Operation at Clustering Nodes of Heterogenous Sensor Networks	75
<i>Ehsan Vossough and Janusz R. Getta</i>	

Facilitating Reuse of Code Checking Rules in Static Code Analysis	91
<i>Vladimir A. Shekhovtsov, Yuriy Tomilko, and Mikhail D. Godlevskiy</i>	
Achieving Adaptivity Through Strategies in a Distributed Software Architecture	103
<i>Claudia Raibulet, Luigi Ubezio, and William Gobbo</i>	
Genetic Algorithm Application for Traffic Light Control	115
<i>Ayad. M. Turkey, M.S. Ahmad, M.Z.M. Yusoff, and N.R. Sabar</i>	
Weaving Business Processes and Rules: A Petri Net Approach	121
<i>Jian Yu, Quan Z. Sheng, Paolo Falcarin, and Maurizio Morisio</i>	

Enterprise Business Process Modelling

Modeling Actions in Dynamic Engineering Design Processes	127
<i>Vadim Ermolayev, Natalya Keberle, Eyck Jentzsch, Richard Sohnius, and Wolf-Ekkehard Matzke</i>	
Replenishment Policy with Deteriorating Raw Material Under a Supply Chain: Complexity and the Use of Ant Colony Optimization	142
<i>Jui-Tsung Wong, Kuei-Hsien Chen, and Chwen-Tzeng Su</i>	
An Algorithm for Propagating-Impact Analysis of Process Evolutions . . .	153
<i>Jeewani Anupama Ginige and Athula Ginige</i>	
Business Process Improvements in Production Planning of ERP System Using Enhanced Process and Integrated Data Models	165
<i>Premaratne Samaranayake</i>	
Computing the Cost of Business Processes	178
<i>Partha Sampath and Martin Wirsing</i>	
Formalizing Computer Forensics Process with UML	184
<i>Chun Ruan and Ewa Huebner</i>	

Information Retrieval and NLP

Improving Product Usage Monitoring and Analysis with Semantic Concepts	190
<i>Mathias Funk, Anne Rozinat, Ana Karla Alves de Medeiros, Piet van der Putten, Henk Corporaal, and Wil van der Aalst</i>	
Using GA and KMP Algorithm to Implement an Approach to Learning Through Intelligent Framework Documentation	202
<i>Hajar Mat Jani and Sai Peck Lee</i>	

Computer-Based Assessment: From Objective Tests to Automated Essay Grading. Now for Automated Essay Writing?	214
<i>Robert Williams and John Nash</i>	

Attitudes Toward ICT of Electronic Distance Education (ePJJ) Students at the Institute of Education Development, University Technology Mara	222
<i>Che Zainab Abdullah, Hashim Ahmad, and Rugayah Hashim</i>	

e-Learning and Training

Metaverse Services: Extensible Learning with Mediated Teleporting into 3D Environments	229
<i>Ioakim Marmaridis and Sharon Griffith</i>	

Combining Simulation and Animation of Queueing Scenarios in a Flash-Based Discrete Event Simulator	240
<i>Ruzelan Khalid, Wolfgang Kreutzer, and Tim Bell</i>	

Preservation of Client Credentials Within Online Multi-user Virtual Environments Used as Learning Spaces	252
<i>Sharon Griffith and Ioakim Marmaridis</i>	

Short Term Stock Prediction Using SOM	262
<i>Prompong Sugunsil and Samerkae Somhom</i>	

Datamining, Datawarehousing, and Visualization

Rules and Patterns for Website Orchestration	268
<i>René Noack</i>	

Visual Intelligence Density	280
<i>Xiaoyan Bai, David White, and David Sundaram</i>	

A Framework for Data Quality in Data Warehousing	292
<i>Rao R. Nemani and Ramesh Konda</i>	

Visuco: A Tool for Visualizing Software Components Usability	298
<i>M^a Ángeles Moraga and Coral Calero</i>	

Information Systems Adaption, Integration, and Security

Study of Using the Meta-model Based Meta-design Paradigm for Developing and Maintaining Web Applications.....	304
<i>Buddhima De Silva and Athula Ginige</i>	

Lost in Translation? Transformation Nets to the Rescue!	315
<i>Manuel Wimmer, Angelika Kusel, Thomas Reiter, Werner Retschitzegger, Wieland Schwinger, and Gerti Kappel</i>	
Metamodeling Foundation for Software and Data Integration	328
<i>Henning Agt, Gregor Bauhoff, Mario Carlsburg, Daniel Kumpe, Ralf Kutsche, and Nikola Milanovic</i>	
Medical Personal Data in Secure Information Systems	340
<i>Tatjana Welzer, Marko Hölbl, Marjan Družovec, Brane Klopčič, Boštjan Brumen, Hannu Jaakkola, and Mirjam Bonačić</i>	
Implementing Medication Management Software Effectively Within a Hospital Environment: Gaining Benefits from Metaphorical Design	346
<i>Salah Awami, Paula M.C. Swatman, and Jean-Pierre Calabretto</i>	

Information Systems Architecture and Technologies

Exploring the Potential of Component-Oriented Software Development Application	355
<i>Hazleen Aris</i>	
On Using Semantic Transformation Algorithms for XML Safe Update	367
<i>Dung Xuan Thi Le and Eric Pardede</i>	
Storing and Querying Graph Data Using Efficient Relational Processing Techniques	379
<i>Sherif Sakr</i>	
SecCom: A Prototype for Integrating Security-Aware Components	393
<i>Khaled M. Khan and Calvin Tan</i>	
Incorporating Software Testing as a Discipline in Curriculum of Computing Courses	404
<i>Simi (Kamini) Bajaj and Shyamala Balram</i>	

Information Systems Applications and Web Intelligence

Intelligent Recruitment Services System	411
<i>Tetyana Shatovska, Victoriya Repka, and Iryna Kamenieva</i>	
Challenges and Opportunities Relating to RFID Implementation in the Healthcare System	420
<i>Belal Chowdhury and Clare D'Souza</i>	
Communities of Practice and Semantic Web Stimulating Collaboration by Document Markup	432
<i>Christine Müller</i>	

eCOMO Papers

Modelling and Analysis Challenges

Conceptualizing Software Life Cycle	438
<i>Sabah S. Al-Fedaghi</i>	
Modeling Complex Adaptive Systems	458
<i>I.T. Hawryszkiewicz</i>	
Designing Modular Architectures for Cross-Organizational Electronic Interaction	469
<i>Christoph Schroth, Beat Schmid, and Willy Müller</i>	

Modelling in Action

Modelling the Bullwhip Effect Dampening Practices in a Limited Capacity Production Network	475
<i>Elena Ciancimino and Salvatore Cannella</i>	
A Comparative Analysis of Knowledge Management in SMEs	487
<i>Maria R. Lee and Yi-Chen Lan</i>	
Supporting Strategic Decision Making in an Enterprise University Through Detecting Patterns of Academic Collaboration	496
<i>Ekta Nankani, Simeon Simoff, Sara Denize, and Louise Young</i>	
Dynamic User Modeling for Personalized Advertisement Delivery on Mobile Devices	508
<i>Luca Paolino, Monica Sebillo, Genoveffa Tortora, Alessandro M. Martellone, David Tacconi, and Giuliana Vitiello</i>	

MBSDI Papers

A Graphical Query Language for Querying Petri Nets	514
<i>Lan Xiao, Li Zheng, Jian Xiao, and Yi Huang</i>	
Model Checking by Generating Observers from an Interface Specification Between Components	526
<i>Tetsuo Hasegawa and Yoshiaki Fukazawa</i>	
A Meta-modeling Framework to Support Accountability in Business Process Modeling	539
<i>Joe Zou, Christopher De Vaney, and Yan Wang</i>	
Extensible and Precise Modeling for Wireless Sensor Networks	551
<i>Bahar Akbal-Delibas, Pruet Boonma, and Junichi Suzuki</i>	
Author Index	563

Model-Based Design - A Chance for Good Quality Products and Services by Integrating Intelligence

András Pataricza

Department of Measurement and Information Systems
Budapest University of Technology and Economics
H-1521 Budapest, Magyar tudósok krt 2/b B420, Hungary
pataric@mit.bme.hu

Abstract. Frequently, model-based computing is looked at as a good paradigm increasing productivity by allowing an increase in the design level, thus in productivity and reusability similar to that what happened several decades ago when software technology changed from machine near assembly programming to high-level languages. However, model-based design (and operation) opens new opportunities to drastically increase the quality of the IT products and the services delivered by them. During the design phase one of the main drivers of the advanced support by the design environment originates in the opportunity of integrating sophisticated mathematics into the design workflow. Formal methods embedded into design workflow may support not only a continuous checking of the conformance to the design rules (which may be specific to the application area by introducing domain specific languages into the design toolchain) but they can support the design process by deliberating the designer from tedious routine tasks. Another option is offer in the form of helping the designer in complex situations by means of integrated multiaspect optimization. The talk will present the new opportunities based on the toolchain developed for safety critical embedded systems at the Budapest University of Technology and OptXware, respectively. At first, it introduces an approach supporting the introduction of domain specific technique in a controlled way into the entire toolchain (covering the creation, model transformation based manipulation, semi-automated synthesis and formal verification). Subsequently, a concept is introduced for the supporting technologies relying on tool integration and information fusion in which the main concepts of service oriented architectures are complemented with a model transformation based semantic model and data integration. Finally, the exploitation of the existence of the requirement and design models for supporting the operation and maintenance phases is addressed.

Bio

Professor András Pataricza received his diploma in Electrical Engineering in 1977 from the Technical University Budapest, his PhD and Doctor of the Hungarian Academy of Sciences degree in 1988 and 2008, respectively. Since 1977 he is with the Department of Measurement and Information Systems. He is the leader of the

Fault-Tolerant Systems Research Group. He was a visiting professor at the University of Erlangen-Nuremberg between 1993 and 1994, and in 2003. He is the author, co-author or editor of 13 books, 7 book chapters, 25 journal articles and 110 conference papers. He has been the Hungarian project leader of several EU projects (HIDE, HIDDENETS, RESIST, SENSORIA, GENESYS and DECOS) and many academic as well as industrial research projects. He was PC member of DSN, EDCC, DDECS, ISAS, FTCS etc. conference series and SC member, general co-chair and program co-chair of EDCC, ISAS and DDECS conference series. He is founder and president of a spin-off company named OptXware established together with the members of his Research Group in order to promote model-based computing with a special emphasis of the aspects and consolidation of dependability.

Process Modelling – What Really Matters

Michael Rosemann

Faculty of Science and Technology, Queensland University of Technology
126 Margaret Street, Brisbane Qld 4000, Australia
m.rosemann@qut.edu.au

Abstract. Process modelling has become one of the most popular forms of conceptual modelling. However, there is an increasing body of evidence suggesting that the requirements of organisations and the focus of related academic research do not sufficiently overlap. This keynote presentation will start with the results of a comparative study on the benefits, issues and challenges as they are perceived by academics, IT vendors and end users. It will become obvious that there is a substantial gap between the priorities of these communities. Recommendations and specific examples will be provided for how to close this gap in order to increase the relevance of research on process modelling. This will cover context-aware process modelling, collaborative process modelling, visualisation and the overall success of process modelling. As a consequence, it will be postulated to be more proactive in terms of collaborations between design-oriented and behavioural researchers.

Bio



Dr Michael Rosemann is a Professor for Information Systems and Co-Leader of the Business Process Management (BPM) Group at Queensland University of Technology, Brisbane, Australia. He is the Chief Investigator of a number of applied research projects funded by the Australian Research Council (ARC) and various industry partners. He was a member of the ARC College of Experts in 2006/07.

A prolific writer, he is the author/editor of seven books, more than 140 refereed papers (incl. MISQ, JAIS, IEEE TKDE, Information Systems) and Editorial Board member of seven international journals. His publications have been translated into German, Russian, Portuguese and Chinese. Michael's PhD students have won the Australian award for the best PhD thesis in Information Systems in 2007 and in 2008. Michael is the co-inventor of seven US patent proposals related to process modelling. Dr Rosemann is the founder and chair of the Australian BPM Community of Practice (bpm-collaboration.com) and he has been the Chair of the 5th International Business Process Management Conference in 2007. He regularly conducts executive training in BPM (www.bpm-training.com) and provided advice to organisations from various industries including telecommunications, banking, insurance, utility, retail, public sector, logistics and film industry.

Robust Web Services Provisioning through On-Demand Replication

Quan Z. Sheng¹, Zakaria Maamar², Jian Yu¹, and Anne H.H. Ngu³

¹ School of Computer Science, The University of Adelaide, SA 5005, Australia
{qsheng, jyu}@cs.adelaide.edu.au

² College of Information Technology, Zayed University, Dubai, UAE
zakaria.maamar@zu.ac.ae

³ Department of Computer Science, Texas State University, San Marcos, TX, USA
angu@txstate.edu

Abstract. Significant difference exists between things that work and things that work well. Availability along with reliability would make Web services the default technology of choice in developing many mission-critical electronic-business applications. Unfortunately, guaranteeing a Web service availability is still a challenge due to the unpredictable number of invocation requests a Web service (e.g., Google Maps) has to handle at a time, as well as the dynamic nature of the Internet (e.g., network disconnections). In addition, the heterogeneity, mobility and distributed nature of Web services makes traditional dependability and availability approaches inappropriate for Web services. In this paper, we describe the design of an on-demand replication approach for robust Web service provisioning. This approach dynamically deploys Web services at appropriate idle hosts, reducing the unavailability of Web services during peak demand for limited computing resources. In addition this approach promotes the decoupling of Web service providers and Web service host providers, which supports a more flexible replication model.

Keywords: Web service, replication, service host, matchmaking.

1 Introduction

The emerging Service-Oriented Computing (SOC) paradigm highlighted by Web services technology promises bringing better interoperability and flexibility to business applications. In order to compete globally, more and more enterprises turn nowadays towards Web services to achieve better visibility and accessibility of their core business competencies (e.g., Amazon Web services¹). Unfortunately, over the past few years, efforts put into SOC were mainly concentrated on making things (e.g., Web services integration) work, but barely on making these things work well [3, 8]. This posed major challenges to enterprises wishing to embrace Web services as a development technology for their IT critical applications.

¹ <http://aws.amazon.com/>

One such challenge, yet still being largely overlooked, is the *availability* (readiness of correct service) and *reliability* (continuity of correct service) of Web services [3,4,8,10]. Given the large number of requests that Web services handle (e.g., Google was searched 4.4 billion times in the US alone in October 2007²), computing hosts upon which these Web services operate can easily turn out to be inadequate for guaranteeing low response-times and unavailability in case of heavy loads. However, enterprises cannot afford interrupting their activities, even for short periods of time. For instance, in April 2002, eBay suffered a 22-hour server-outage affecting most of its online auctions, costing five million US dollars lost in revenue and over five billion dollars in market capitalization.

A recent emerging trend for solving the high-availability issue of Web services is centered around *replication* [4,10]. The underlying idea is to spread replicas over various computing hosts and direct invocation requests to appropriate hosts (e.g., with lower workload). The aim of the work reported in this paper is to enhance the fundamental understanding of robust Web services provisioning and to develop a novel approach by advancing the current state of art in this direction. Our main contributions are summarized as follows:

- Web services are deployed on-demand by targeting appropriate idle computing hosts. This reduces the unavailability risk and achieves better load-balancing. A core architectural design in this deployment is the clear separation between *Web service* providers and *Web service host* providers.
- Dynamic Web services replication model that enables proactive deployment of replicas so that a certain desired availability level is maintained. This model helps determine how many replicas are needed, when and where they should be created and deployed. The deployment of replicas remains transparent to clients so that they are unnecessarily aware of the underlying replication.
- Mechanisms for Web service hosts matchmaking and selection. A *matchmaking* process and a *multi-criteria utility* function are introduced to dynamically select appropriate computing hosts based on Web services' requirements.

The remainder of the paper is organized as follows. Section 2 overviews the basic concepts that underpin the robust Web service provisioning approach and Section 3 introduces the system design. Section 4 describes the solution on dynamic replication of Web services. Section 5 introduces our algorithm for service host matchmaking. Implementation and some experimental results are documented in Section 6. Finally, Section 7 concludes the paper.

2 Preliminaries

This section introduces some basic concepts upon which our robust Web services provisioning approach is built namely, Web services anatomy, Web service hosting systems, and service code mobility.

² <http://www.buzzmachine.com/2007/12/29/google-is-god/>

2.1 Anatomy of Web Services

Interacting with Web services relies on exchanging SOAP messages. These messages contain various details such as input/output data format that are specified and included in WSDL documents. Figure 1 depicts a typical interaction-session with a Web service. Initially, a provider informs potential clients of the address (i.e., endpoint) that its Web service uses to receive SOAP messages. This address is posted on a registry such as UDDI. Upon reception, a message is converted by the Web service's processing logic into a form that is appropriate for the computing system upon which this Web service runs.

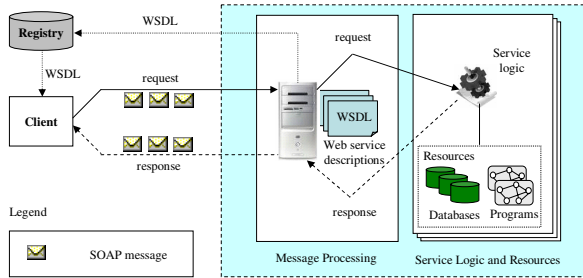


Fig. 1. Structure of a typical interaction-session with a Web service

The separation of concerns on business logic and message handling, which is actually a fundamental property of the concept of Web service, offers several advantages: i) free evolution of Web services (e.g., performance improvement) without affecting binding mechanisms of service consumers, and ii) free decoupling of Web service's endpoints (i.e., locations where SOAP messages are sent) from Web service invocation (i.e., locations where the Web service is executed).

2.2 Web Services and Web Service Hosts

Inspired by the Web site hosting trend where business (and personal) Web pages are stored in one or multiple hosting service providers (e.g., [HostMonster³](http://www.hostmonster.com/)), we advocate the separation of *Web services* providers and *host* providers.

A Web service implements the functionalities of an application, developed by a particular provider. This provider is also responsible for advertising the Web service to potential customers and installing it on appropriate hosts, either local or distant. A Web service can be concurrently made available through replicas on distinct service hosts. To deal with service hosts variety (e.g., Windows, Linux and Solaris), a Web service provider needs to maintain ready-to-deploy versions of this Web service according to the hardware and software requirements of these hosts. A deployable version is typically a Web service bundle that contains all what is necessary to deploy the service in a single file (e.g., war or jar file).

³ <http://www.hostmonster.com/>

A host provider has the authority to control the computing resources (e.g., a set of servers installed Apache Axis) it manages for the sake of accommodating the execution needs of Web services. It accepts SOAP messages sent from a Web service provider, routes these messages to the host where the corresponding Web service is located, and returns Web service results (if any), in another SOAP message, to the Web service provider.

2.3 Stationary and Mobile Web Services

It is accepted that code mobility is suitable for load balancing, performance optimization, and disconnection handling [5]. Code mobility would significantly contribute to secure prompt responses; applications running on heavy loaded or unavailable hosts can be dynamically migrated to other better hosts. Examples of mobile code technologies include Aglets⁴, Java, Sumatra [1], and μ Code⁵.

Injecting code mobility into Web services sheds the light on *stationary* and *mobile* Web services. Stationary Web services are location-dependent; they can not move because for various reasons such as resource availability (e.g., a specific database) on a dedicated host. On top of behaving like their counterparts in terms of accepting invocation requests and taking part in composition scenarios, mobile Web services are i) location-independent, ii) enhanced with migration capabilities that make them move to distant hosts for execution, and iii) stateless. As a result, a mobile Web service can be executed on any arbitrary host as long as this host accommodates this mobile Web service's execution needs.

Mobile Web services have already attracted the attention of the research community as the large number of references witness [2,6,7]. A simple way is to pack the bundle of a mobile Web service into a single file (e.g., war file) that can be dynamically deployed at appropriate service hosts. In [2], mobile Web services are considered as synthesis of Web services and mobile agents. In [7], Liu and Lewis develop an XML-based mobile code language called X# and presents an approach for enabling Web services containers to accept and run mobile codes. It should be noted that proposing techniques for the development of mobile Web services is not the focus of this paper. Instead, we adopt some of the strategies used in mobile Web services in our approach for dynamic replication of Web services that can move around.

3 System Design

Central to our approach for making Web services provisioning robust are the following components (Figure 2): *Web service provider*, *Web service manager*, pool of *Web service hosts*, and *matchmaker*.

The Web service provider develops Web services and stores their deployable versions in a Web service code-repository. Each Web service provider is associated with a Web service manager that consists of four components, namely

⁴ <http://www.trl.ibm.com/aglets/>

⁵ <http://mucode.sourceforge.net/>

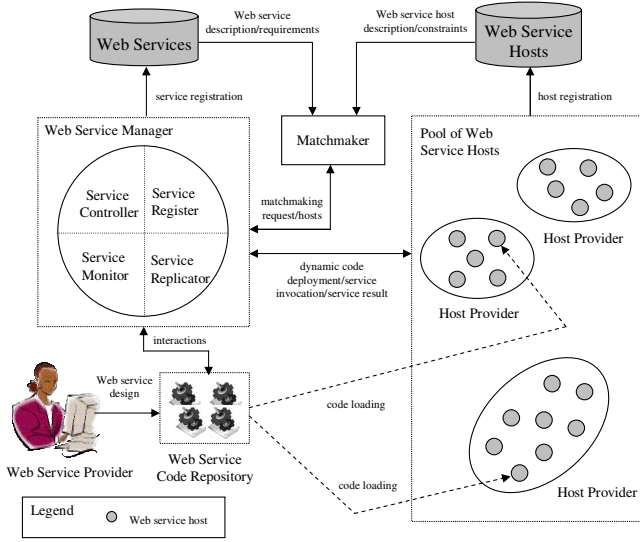


Fig. 2. System architecture

service controller, *service replicator*, *service register*, and *service monitor*, which collaborate together to achieve and sustain Web services high-availability.

The service monitor oversees the execution status of a Web service (e.g., exceptions). The service controller interacts with the service monitor and coordinates the execution of the corresponding Web service. If the value of a monitored item (e.g., CPU usage) is beyond a critical point—which can be set by the service provider—and the Web service is movable/mobile [2], the controller invites the service replicator to move the Web service to a selected Web service host for execution. If such a Web service host does not exist or the Web service is stationary, the controller triggers an exception handling policy (e.g., forward the Web service invocation to a substitute peer), if such a policy has been specified by the service provider. Different solutions to Web services substitution are reported in the literature such as [9].

The service replicator is a light weight scheduler that helps the controller execute the associated Web service on other computing resources when necessary. The replicator decides i) the number of replicas needed for a desired availability degree, and ii) when and where the replicas are deployed. Web service codes are transferred to hosts and invocation requests are dynamically routed to these Web services' new locations. Its underlying technique is what we call *dynamic service replication* and more details are reported in Section 4.

Each host provider controls a cluster of Web service hosts. These hosts have monitoring modules (not shown in the figure for the sake of simplicity) that oversee resource consumption in terms of CPU, memory, etc. Web services and Web service hosts can register themselves using public registries like UDDI.

To this end, an appropriate tModel⁶ (e.g., `ServiceHost`) to describe hosts is required so that Web services can locate them. Service host selection is another important factor to the success of our approach. The selection of service hosts is based on a matchmaking mechanism that is implemented in the matchmaker. More details on host matchmaking are given in Section 5.

4 Dynamic Web Services Replication

Replication is well-known to improve system availability. In this section, we introduce a mechanism that permits to proactively create Web service replicas so that the availability requirement is sustained⁷. This mechanism relies on a *replication decision model* that helps a Web service determine how many replicas are needed, and when and where they should be created and deployed.

4.1 Replication Decision Model

In replication there is always a trade-off between availability and cost. On the one hand, more hosts accepting a Web service means better and higher availability. On the other hand, more replicas at various hosts means higher overhead (e.g., resource consumption such as bandwidth and storage space). It is, therefore, essential to have a replication decision model, which helps the service replicator determine i) the optimal number of the replicas in order to meet a Web service's availability requirement, and ii) time and locations for deploying the new replicas.

Calculating the Number of Replicas. Given the failure probability of each Web service host p , total number of replicas $\mathcal{N}_t$ of a Web service s , and availability threshold $\mathcal{A}$ of s , the following formula must be satisfied,

$$\mathcal{A}(s) \leq 1 - p^{\mathcal{N}_t(s)} \quad (1)$$

where $p^{\mathcal{N}_t(s)}$ represents the probability of all replicas of Web service s being unavailable and $1 - p^{\mathcal{N}_t(s)}$ represents the probability of at least one replica of s being available. The meanings of the notations are given in Table 1. From Formula 1, we can easily get

$$\mathcal{N}_t(s) \geq \log_p^{1-\mathcal{A}(s)} \quad (2)$$

to calculate the right number of replicas for a certain availability threshold. For example, assume that the failure probability of service hosts p is 10% (low). To satisfy the availability threshold of 99.99% of a Web service, 4 replicas are recommended. If p is 50% (medium), the number of replicas is now 14. If p becomes 90% (high), this number increases to 88.

⁶ In UDDI, a tModel provides a semantic classification of the functionality of a service or a host, together with a formal description of its interfaces.

⁷ In the remainder, we assume that the Web service is mobile when Web service replication is referred to.

Table 1. Notations and their meanings

Notation	Meaning
$\mathcal{A}(s)$	Required availability threshold of Web service s
p	Failure probability of each service host
$\mathcal{N}_t(s)$	Total number of replicas needed for the Web service s
$\mathcal{N}_e(s)$	Total number of replicas currently existed for the Web service s

Deploying Replicas. Once the number of required replicas (i.e., $\mathcal{N}_t$) of a Web service s is known, the service replicator of s might decide to deploy additional replicas of s on top of the existing ones (i.e., $\mathcal{N}_e$). The service replicator takes no action if $\mathcal{N}_t(s)$ is less than or equal to $\mathcal{N}_e(s)$. Otherwise, the replicator has to deploy $\mathcal{N}_t(s) - \mathcal{N}_e(s)$ replicas of s to remote hosts. This typically involves the following tasks: i) selecting suitable service hosts for the replicas (Section 5), and ii) deploying replicas to the selected service hosts. A simple way for obtaining the value of $\mathcal{N}_e$ is to count the Web service hosts that accepted replicas of the Web service, which is maintained in the *routing table* to be detailed in Section 4.2.

Due to dynamic nature of the Internet (e.g., retraction of a service host without prior notice), an interesting problem is that when a Web service should check whether it needs to deploy more replicas in order to meet the required availability level. We propose that the service replicator should periodically compare $\mathcal{N}_t(s)$ with $\mathcal{N}_e(s)$. The periodicity of the checks depends on the availability level of the Web service. In case of critical availability, a shorter frequency is applied, which means more frequent checks. Otherwise, a longer frequency is applied. The service replicator can also consider the checking history. For example, if there are no actions needed during the last few (e.g., 3) checks, the replicator can increase the check interval. Contrarily, if new replicas were needed in the last few checks, the replicator would decrease the check interval.

4.2 Deployment Transparency

Transparency is an important issue from consumers' point of view. They should by no means be aware of the existence of replicas. In our design, a provider registers a Web service in a UDDI repository by making the service controller (Figure 2) act as this Web service's endpoint. It is the service controller's responsibility to receive invocation messages and forward them to the appropriate service hosts for actual service invocation. This is achieved with the help of a *routing table* that the service replicator maintains. This table keeps the access information on the deployed copies by simply recording a pair of $\langle \text{Host-ID}, \text{EndPoint} \rangle$ where **Host-ID** is the identifier of the service host upon which the Web service is deployed and **EndPoint** is the access point of the deployed replica.

It should be noted that a replica deployed on a host may not be working due to reasons that could be related to end of contract or giving room to higher priority Web services. For the sake of simplicity, we assume that the routing table keeps updated information. For example, if a deployed Web service is removed from a service host, its record will be removed from the routing table as

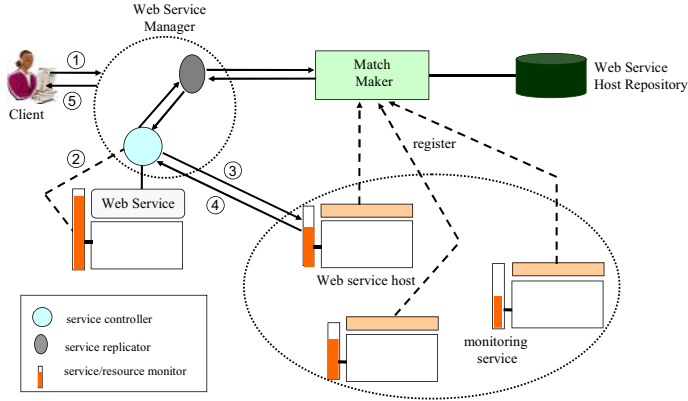


Fig. 3. Interactions of system components for Web service invocation

well. When the need for service invocation on a service host rises, the service controller consults the routing table and forwards (routes) the service invocation message to the other host(s) for processing. Strategies that the controller can apply include: i) selecting the best service host from the routing table for invocation, and ii) selecting all the service hosts for invocation and returning the first response to the customer.

Figure 3 illustrates the high level interactions among the components during Web service invocation. When a client invokes a Web service (step 1), the service controller⁸ contacts the service monitor about the status of the local service host (step 2). If the host is overloaded (e.g., CPU usage of the host is higher than 90%), the controller picks up a service host, by consulting with the service replicator and the matchmaker and forwards the invocation message to the endpoint of the corresponding replica (step 3). The service result is then returned to the controller (step 4). Finally, the Web service manager returns the result to the client (step 5).

5 Host Matchmaking and Selection

The approach we have presented relies on a *matchmaking* mechanism to locate and select appropriate Web service hosts. The basic idea of host matchmaking is summarized as follows: service replicators and host providers advertise requirements and characteristics of Web services and service hosts; a designated module (i.e., *matchmaker*) matches the advertisements in a manner that meets the requirements and characteristics specified in the respective advertisements.

The descriptions of Web services and resources consist of two parts: *attributes* and *constraints*. The attributes for a Web service includes characteristics such as service location, mobility, and input/output parameters. The attributes for

⁸ For the sake of simplicity, other components in the Web service manager are excluded from Figure 3.

a host include properties such as CPU usage, free memory, and price. The constraint part includes limitations defined by the providers of Web services and hosts. For example, a host provider may specify that it will not service any request coming from company *A* due to regular payment delays. Similarly, the provider of a (mobile) Web service may specify that only hosts with more than 200K bytes of free disk space and at least 128K bytes of free memory are eligible to host this Web service. Service hosts can be described using W3C's RDF (Resource Description Framework)⁹, while Web services can be described using WSDL (Web Service Description Language)¹⁰. We do not give detailed description of RDF and WSDL due to space limitations.

Matchmaking Interactions. Matchmaking is defined as a process that requires a service description as input and returns a set of potential hosts. A host is matched with a service if both the requirement expressions of the service and the constraints of the host are evaluated to true.

Several steps involve in the interactions of the matchmaking process. Service replicators and host providers develop descriptions for their requirements and attributes and send them to the matchmaker (step 1). The matchmaker then, invokes a *matchmaking algorithm* by which matches are identified (step 2). The invocation includes finding service-host description pairs that satisfy the constraints and requirements of hosts and services. We will detail this step later. After the matching step, the matchmaker notifies the service replicator and the host providers (step 3). The service replicator and the host provider(s) then contact each other and establish a working relationship (step 4). It should be noted that a matched host of a service does not mean that the host is allocated to the Web service. Rather, the matching is a mutual introduction between Web services and hosts and the real working relationship can be consequently built after the communication between the two parts.

Expression Evaluation. Expression evaluation plays an important role in the matchmaking process. To evaluate a constraint expression on a service description, the attribute of the expression is replaced with the value of the corresponding attribute of the host. If the corresponding attribute does not exist in the host description, the attribute of the expression is replaced with the constant **undefined**. In our matchmaking algorithm, expressions containing **undefined** are eventually evaluated to *false*. The constraints of the host descriptions have the similar evaluation process. When receiving a request from a service replicator, the matchmaker takes the Web service description, evaluates all the hosts advertised in the host repository using the matchmaking algorithm described above, and returns a set of matched hosts to the service replicator.

For example, assume that a Web service needs at least 128K bytes free memory to run (i.e., `memoryfree` $\geq$ 128K). The matchmaker scans the description of a host for the attribute `memoryfree`. The value of the attribute (e.g., 512000K

⁹ <http://www.w3.org/TR/REC-rdf-syntax>

¹⁰ <http://www.w3.org/TR/wsdl>

bytes) is used to replace the attribute in the expression (i.e., $512000 \geq 128$), which is in turn evaluated to *true* by the matchmaker.

Hosts Selection. For a specific Web service, there could be multiple potential hosts matched for executing the Web service. The service provider, therefore, should be able to choose the best host (or top N best hosts) that satisfies its particular needs from the matched hosts. To specify preferences over hosts of a particular Web service, we exploit a *multi-criteria utility function*,

$$\mathcal{U}(h) = \sum_{i \in \mathcal{SA}} w_i \cdot \text{Score}_i(h) \quad (3)$$

where i) h is a host, ii) $\text{Score}_i(h)$ is an attribute scoring function, iii) $\mathcal{SA}$ is the set of selection attributes, and iv) w_i is the weight assigned to attribute i . The scoring service computes the weighted sum of criteria scores using the weight property of each selection criterion. It selects the host that produces the higher overall score according to the multi-criteria utility function. Several criteria—such as *price*, *availability*, *reliability*, and *reputation*—can be used in the function. Due to the space constraint, the criteria calculation is not described in this paper. Interested readers are referred to [11] for details.

6 Implementation and Experimentation

6.1 Implementation

This section describes the system implementation with focus on the matchmaker and the Web service manager (Figure 2).

The Matchmaker. Two repositories are included in the system, namely *Web service repository* and *Web service host repository*. Both repositories are implemented as UDDI registries. Each host is represented as an RDF document. A separate tModel of type `hostSpec` is created per Web service host, including a tModel key, a name (i.e., host name), an optional description, and a URL that points to the location of the host description document. WSDL is used to specify Web services. Since WSDL focuses on how to invoke a Web service, some of the attributes (e.g., stationary or mobile service) proposed in our approach are not supported. To overcome this limitation, such attributes are specified as tModels. The keys of these tModels are included in the `categoryBag` of the tModel of a Web service so that the Web service knows the descriptions of these attributes.

The matchmaker is implemented using the UDDI Java API (UDDI4J) provided by IBM Web Services Toolkit 2.4 (WSTK). This matchmaker provides two kinds of interfaces for both repositories, including an *advertise interface* and a *search interface*. The former interface is used to publish Web services and service hosts, while the latter interface is used to search service hosts using the approach presented in this paper.

The Web Service Manager. The functionalities of the service controller are realized by a pre-built Java class, called **Controller**, which provides an operation called **coordinate** to receive invocation messages, manage service instances (e.g., initializing an instance in a remote host with a replica of the Web service), trace service invocations, and trigger service replication actions.

The controller relies on the service monitor and the service replicator to make intelligent decision on when and where to perform the replication. In particular, the service replicator is the module that maintains a desired availability degree of a Web service. It provides a specific method called **replication** that can be used to perform replica number calculation and replica deployment. In addition, the service replicator maintains a routing table that contains the information on the replicas of a Web service (e.g., endpoints of the replicas). Currently, the routing table is simply implemented as an XML document.

6.2 Experimentation

In this section, we present some preliminary experimental results with emphasis on evaluating the performance overhead introduced by our approach. We implemented a simple Web service that takes a string *str* as input and outputs another string which simply joins **Hello** and *str* together. The service is deployed using Apache Axis. Four machines were used in the experiments. One machine was dedicated to the original Web service (including the Web service manager) and three were used to host replicas of this Web service. All machines have the same configuration of Pentium III 933MHz and 512Mb RAM, and are connected to a LAN through 100Mbps/sec Ethernet cards. Each machine runs Apache Axis 1.2 with Jakarta Tomcat 5.5, Debian Linux, and the Java 2 Standard Edition V1.4.2.

To evaluate the performance overhead of our approach, we considered two cases: i) invoking the string-joining Web service directly (i.e., non-replication case where service requests are sent to the Web service), and ii) invoking the string-joining Web service using our approach (i.e., replication case where service requests were sent to the service controller of the Web service). We simulated 10

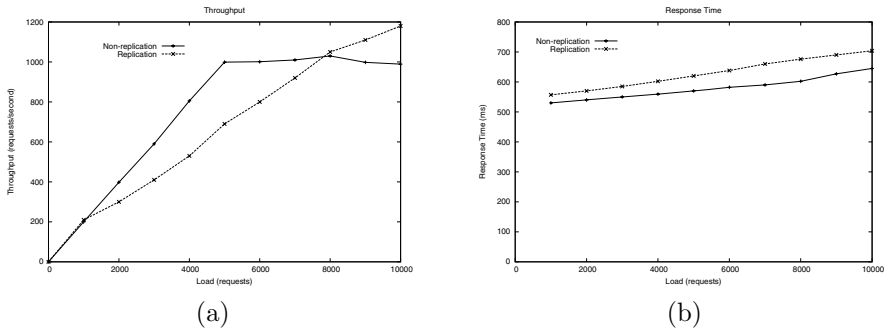


Fig. 4. Performance study: (a) throughput, (b) response time

clients, in another machine, which submitted invocation requests simultaneously to both cases respectively for service execution. The number of request messages ranges from 1,000 to 10,000.

Figure 4(a) gives the average throughput of the two cases. We can see that the throughput of the non-replication case reaches its maximum around 5,000 requests. After that, the throughput remains steady and does not increase anymore. The throughput of the replication case, on the other hand, keeps increasing even when the number of requests reaches 10,000. Figure 4(b) shows the average response time. We can see that there is a relatively small overhead in the replication case. For example, when the number of request messages reaches 6,000, the average response time of non-replication case is 582ms, whereas for the replication case, it is 638ms. This overhead is reasonable due to the additional method calls that are inherent to our Web service manager implementation.

7 Conclusion

With rapid adoption of Service-Oriented Computing (SOC), more and more organizations are deploying their business competencies as Web services over the Internet. In these scenarios, service utility is often determined by its availability rather than the traditional metric of raw performance. Unfortunately, guaranteeing a Web service availability is still a challenge due to the unpredictable number of invocation requests a Web service has to handle at a time, as well as the dynamic nature of the Web. In this paper, we presented an approach that exploits service replication to achieve robust Web services provisioning. The main features of this approach are: i) a dynamic Web service replication model to proactively deploy replicas of a Web service for a desired availability level, and ii) mechanisms for Web service hosts matchmaking and selection. The preliminary results show that the cost of high availability (i.e., overheads) is acceptable. The results also show that our approach achieves a better throughput.

Our ongoing work includes further performance study on the on-demand replication strategy (e.g., in a WAN environment) of the proposed techniques. This includes simulating more complex composite services with some component services which are location dependence. It also requires more extensive study on composite services which have services that can be executed in parallel. In addition, we plan to look into leveraging the dynamic information of service hosts (e.g., amount of free memory) and Web services (e.g., quality of service) for the matching and selecting of the best host.

References

1. Acharya, A., Ranganathan, M., Saltz, J.: Sumatra: A Language for Resource-Aware Mobile Programs. In: Tschudin, C.F., Vitek, J. (eds.) MOS 1996. LNCS, vol. 1222, pp. 111–130. Springer, Heidelberg (1997)
2. Adacal, M., Bener, A.B.: Mobile Web Services: A New Agent-Based Framework. IEEE Internet Computing 10(3), 58–65 (2006)

3. Birman, K., van Renesse, R., Vogels, W.: Adding High Availability and Autonomic Behavior to Web Services. In: Proc. of the 26th Intl. Conf. on Software Engineering (ICSE 2004), Scotland, UK (May 2004)
4. Chan, P., Lyu, M., Malek, M.: Reliable Web Services: Methodology, Experiment and Modeling. In: Proc. of IEEE Intl. Conf. on Web Services (ICWS 2007), Utah, USA (July 2007)
5. Fuggetta, A., Picco, G., Vigna, G.: Understanding Code Mobility. *IEEE Trans. on Software Engineering* 24(5), 342–361 (1998)
6. Hirsch, F., Kemp, J., Ilkka, J.: *Mobile Web Services: Architecture and Implementation*. John Wiley & Sons, Chichester (2006)
7. Liu, P., Lewis, M.: Mobile Code Enabled Web Services. In: Proc. of IEEE Intl. Conf. on Web Services (ICWS 2005), Orlando FL, USA (July 2005)
8. Papazoglou, M., Traverso, P., Dustdar, S., Leymann, F.: Service-Oriented Computing: State of the Art and Research Challenges. *Computer* 40(11), 38–45 (2007)
9. Ribeiro, J., do Carmo, G., Valente, M., Mendonça, N.: Smart Proxies for Accessing Replicated Web Services. *IEEE Distributed Systems Online* 8(12) (2007)
10. Salas, J., Pérez-Sorrosal, F., Patiño-Martínez, M., Jiménez-Peris, R.: WS-Replication: A Framework for Highly Available Web Services. In: Proc. of the 15th Intl. World Wide Web Conf. (WWW 2006), Edinburgh, Scotland (May 2006)
11. Zeng, L., Benatallah, B., Dumas, M., Kalagnanam, J., Sheng, Q.Z.: Quality Driven Web Services Composition. In: Proc. of the 12th Intl. World Wide Web Conf. (WWW 2003), Budapest, Hungary (2003)

Modelling and Maintenance of Very Large Database Schemata Using Meta-structures

Hui Ma¹, Klaus-Dieter Schewe², and Bernhard Thalheim³

¹ Victoria University of Wellington, School of Engineering and Computer Science,
Wellington, New Zealand

`hui.ma@ecs.vuw.ac.nz`

² Information Science Research Centre, Palmerston North, New Zealand

`kdschewe@acm.org`

³ Christian-Albrechts-University Kiel, Institute of Computer Science, Kiel, Germany

`thalheim@is.informatik.uni-kiel.de`

Abstract. Practical experience shows that the maintenance of databases with a very large schema causes severe problems, and no systematic support is provided. In this paper we address this problem. Based on the analysis of a large number of very large database schemata we identify twelve frequently recurring meta-structures in three categories associated with schema construction, lifespan and context. We argue that systematic use of these meta-structures will ease the modelling and maintenance of very large database schemata.

Keywords: Database modelling, schema maintenance, meta-structure.

1 Introduction

While data modellers learn about data modelling by means of small “toy” examples, the database schemata that are developed in practical projects tend to become very large. For instance, the relational SAP/R3 schema contains more than 21,000 tables. Moody discovered that as soon as ER schemata exceed 20 entity- and relationship types, they already become hard to read and comprehend for many developers [7].

Therefore, the common observation that very large database schemata are error-prone, hard to read and consequently difficult to maintain is not surprising at all. Common problems comprise repeated components as e.g. in the LH Cargo database schema with respect to transport data or in the SAP/R3 schema with respect to addresses.

Some remedies to the problem have already been discussed in previous work of some of the authors, and applied in some database development projects. For instance, modular techniques such as *design by units* [13] allow schemata to be drastically simplified by exploiting principles of hiding and encapsulation that are known from Software Engineering. Different subschemata are connected by bridge types. *Component engineering* [9] extends this approach by means of

view-centered components with well-defined composition operators, and *hierarchy abstraction* [16] permits to model objects on various levels of detail.

In order to contribute to a systematic development of very large schemata the *co-design* approach, which integrates structure, functionality and interactivity modelling, emphasises the initial modelling of skeletons of components, which is then subject to further refinement [17]. Thus, components representing subschemata form the building blocks, and they are integrated in skeleton schemata by means of connector types, which commonly are modelled by relationship types.

In this paper we further develop the method for systematic schema development focussing on very large schemata. We first analyse skeletons and subschemata more deeply in Section 2 and identify distinguishing dimensions [3]. In Section 3, based on the analysis of more than 8500 database schemata, of which around 3500 should be considered very large we identify twelve frequently recurring meta-structures, which determine the skeleton schema. These meta-structures are classified into three categories addressing schema construction, lifespan and context. Finally, in Section 4 we elaborate more on the application of meta-structures in data modelling, but due to space restrictions some formal details have to be outsourced. In a concurrent submission [6] we elaborate on the handling of the identified meta-structures in a more formal way.

2 Internal Dimensions of Skeletons and Subschemata

A *component* – formally defined in [9,16] – is a database schema together with import and export interfaces for connecting it to other components by standardised interface techniques. *Schema skeletons* [15] provide a framework for the general architecture of an application, to which details such as types are to be added. They are composed of *units*, which are defined by sets of components provided this set can be semantically separated from all other components without losing application information. Units may contain entity, relationship and cluster types, and the types in it should have a certain affinity or adhesion to each other.

In addition, units may be associated with each other in a variety of ways reflecting the general associations within an application. Associations group the relation of units by their meaning. Therefore, different associations may exist between the same units. Associations can also relate associations with each other. Therefore, structuring mechanisms as provided by the higher-order entity-relationship model [13] may be used to describe skeletons.

The usage of types in a database schema differs in many aspects. In order to support the maintenance of very large schemata this diversity of usage should be made explicit. Following an analysis of usage patterns [9] leads to a number of internal dimensions including the following important ones:

- Types may be specialized on the basis of roles objects play or categories into which objects are separated. This *specialization dimension* usually leads to

subtype, role, and categorisation hierarchies, and to versions for development, representation or measures.

- As objects in the application domain hardly ever occur in isolation, we are interested in representing their associations by bridging related types, and adding meta-characterisation on data quality. This *association dimension* often addresses specific facets of an application such as points of view, application areas, and workflows that can be separated from each other.
- Data may be integrated into complex objects at runtime, and links to business steps and rules as well as log, history and usage information may be stored. Furthermore, meta-properties may be associated with objects such as category, source and quality information. This defines the *usage, meta-characterisation* or *log dimension*. Dockets [10] may be used for tracking processing information, superimposed schemata for explicit log of the treatment of the objects, and provenance schemata for the injection of meta-schemata.
- As data usage is often restricted to some user roles, there is a *rights and obligations dimension*, which entails that the characterisation of user activities is often enfolded into the schema.
- As data varies over time and different facets are needed at different moments, there is a *data quality, lifespan and history dimension* for modelling data history and quality, e.g. source data, and data referring to the business process, source restrictions, quality parameters etc. With respect to time the dimension distinguishes between transaction time, user-defined time, validity time, and availability time.
- The *meta-data dimension* refers to temporal, spatial, ownership, representation or context data that is often associated with core data. These meta-data are typically added after the core data has been obtained.

We often observe that very large database schemata incorporate some or all of these dimensions, which explains the difficulty for reading and comprehension. For instance, various architectures such as technical and application architecture may co-appear within a schema [11].

Furthermore, during its lifetime a database schema, which may originally have captured just the normalised structure of the application domain, is subjected to performance considerations and extended in various ways by views. A typical example for a complete schema full of derived data is given by OLAP applications [5]. Thus, at each stage the full schema is in fact the result of folding extensions by means of a so-called *grounding schema* into the core database schema.

3 Meta-structures in Subschemas and Schema Skeletons

Based on an extensive study of a large number of conceptual database schemata we identify frequently occurring meta-structures and classify them in three categories according to construction, lifespan and context. In the following we describe these meta-structures. Due to space restrictions the description will only contain sufficient details for some of the meta-structures, whereas for the others it will necessarily be rather terse.

3.1 Construction Meta-structures

Structures are based on building blocks such as attributes, entity types and relationship types. In order to capture also versions, variations, specialisations, application restrictions, etc. structures can become rather complex. As observed in [9] complex structures can be primarily described on the basis of *star* and *snowflake meta-structures*. In addition, *bulk meta-structures* describing the similarity between things and thus enable generalisation and combination, and *architecture meta-structures* describe the internal construction by building blocks and the interfaces between them.

Star and Snowflake Meta-Structures. Star typing has been used already for a long time outside the database community. The star constructor permits to construct associations within systems that are characterized by complex branching, diversification and distribution alternatives. Such structures appear in a number of situations such as composition and consolidation, complex branching analysis and decision support systems.

A *star meta-structure* is characterized by a core entity (or relationship) type used for storing basic data, and a number of subtypes of the entity type that are used to capture additional properties [16]. A typical star structure is shown in Figure 1 for the **Address** entity type. In the same fashion a *snowflake schema*, the one in Figure 2 – shown without attributes – represents the information structure of documented contributions of members of working groups during certain time periods.

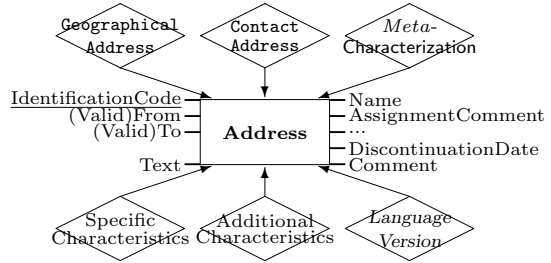


Fig. 1. The General Structure of Addresses

Bulk Meta-Structures. Types used in schemata in a very similar way can be clustered together on the basis of a classification.

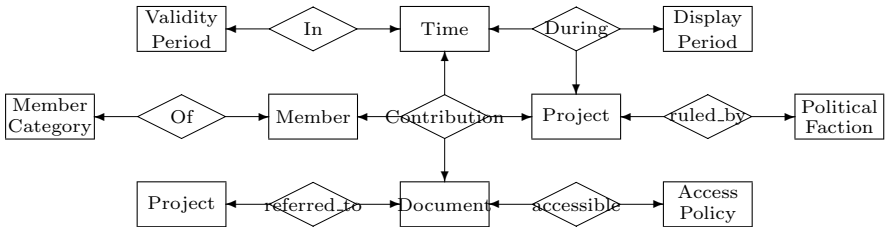


Fig. 2. Snowflake Schema on Contributions

Let us exemplify this generalisation approach for the commenting process in an e-community application. The relationship types **Made**, **Commented**, and **Reused** in Figure 3 are all similar. They associate contributions with both **Group** and **Person**. They are used together and at the same objects, i.e. each contribution object is at the same time associated with one group and one person. We can combine the three relationship types into the type **ContributionAssociation** as shown in Figure 4. The type **ContributionAssociationClassifier** and the domain $\{\text{Made}, \text{Commented}, \text{Reused}\}$ for the attribute **ContractionDomain** can be used to reconstruct the three original relationship types. The handling of classes that are bound by the same behaviour and occurrence can be simplified by this construction.

In general, the meta-structure can be described as follows:

Assume to be given a *central type* C and other types that are associated with C by a set of relationship types $\{A_1, \dots, A_n\}$ by means of an *occurrence frame* F . The occurrence frame can be such that either all inclusion constraints $A_i[C] \subseteq A_j[C]$ for $1 \leq i, j \leq n$ must hold or another set of inclusion constraints.

Now we combine the types $\{A_1, \dots, A_n\}$ into a type **BulkType** with an additional component **ContractionAssistant**, and attributes **Identif** to identify objects of this type, **ContractionDomain** with domain $\{A_1, \dots, A_n\}$, and **ContractionBinder** with domain F . This is shown in Figure 5.

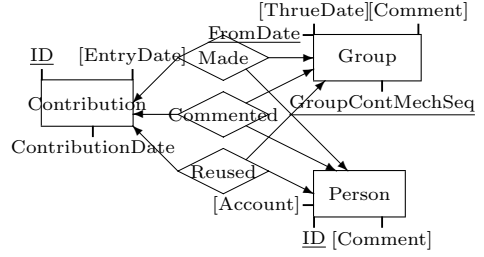


Fig. 3. E-Community Application

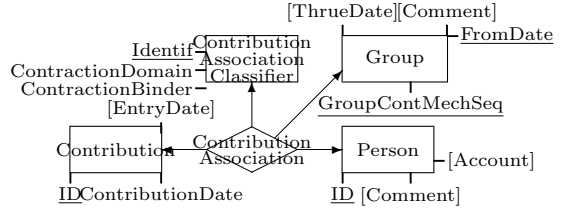


Fig. 4. Bulk Meta-Structure for E-Community

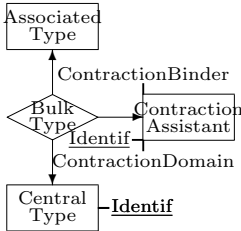


Fig. 5. General Bulk Meta-Structure

Architecture and Constructor-Based Meta-Structures. Categorisation and compartmentalization have been widely used for modelling complex structures. For instance, the architecture of SAP R/3 has often been displayed in form of a waffle. Therefore, we adopt the term *waffle meta-structure* or *architecture meta-structure* for structures that arise this way. These meta-structures are especially useful for the modelling of distributed systems with local components and behaviour. They provide solutions for interface management, replication, encapsulation and inheritance,

and are predominant in component-based development and data warehouse modelling.

Star and snowflake schemata may be composed by composition operators such as *product*, *nest*, *disjoint union*, *difference* and *powerset*. These operators permit the construction of any schema of interest, as they are complete for sets. A structural approach as in [1] can be employed. Thus, all constructors known for database schemata may also be applied to meta-schema construction.

3.2 Lifespan Meta-structures

The evolution of an application over its lifetime is orthogonal to the construction. This leads to a number of *lifespan meta-structures*, which we describe next. *Evolution meta-structures* record life stages similar to workflows, *circulation meta-structures* display the phases in the lifespan of objects, *incremental meta-structures* permit the recording of the development, enhancement and ageing of objects, *loop meta-structures* support chaining and scaling to different perspectives of objects, and *network meta-structures* permit the flexible treatment of objects during their evolution by supporting to pass objects in a variety of evolution paths and enable multi-object collaboration.

Evolution Meta-Structures. By using a *flow* constructor evolution meta-structures permit the construction of a well-communicating set of types with a P2P data exchange among the associated types. Such associations often appear in workflow applications, business processes, customer scenarios, and when identifying variances. Evolution is based on the treatment of *stages* of objects. Objects are passed to handling agents (teams), which maintain and update their specific properties.

Circulation Meta-Structures. Objects may be related to each other by life-cycle stages such as repetition, self-reinforcement and self-correction. Typical examples are objects representing iterative processes, recurring phenomena

or time-dependent activities. A circulation meta-structure supports primarily iterative processes.

Circulation meta-structures permit to display objects in different phases. For instance, legal document handling in the SeSAM system, an e-government application, is based on such phases:

DocumentForm, ProposedDocument, DocumentInReviewing, AcceptedDocument,

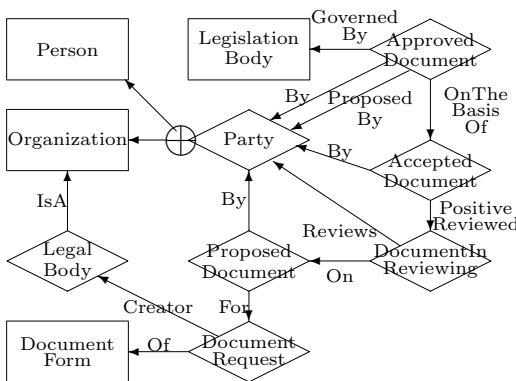


Fig. 6. Incremental Meta-Structure

`RejectedDocument`, `FinalVersionDocument`, `ApprovedDocument`, and `ArchivedDocument`. The circulation model is supported by a phase-based dynamic semantics [13]. Alternatively, an incremental meta-structure could be chosen as shown in Figure 6.

Incremental Meta-Structures. Incremental meta-structures enable the production of new associations based on a core object. It employs containment, sharing of common properties or resources, and alternatives. Typical examples are found in applications, in which processes collect a range of inputs, generate multiple outcomes, or create multiple designs.

Incremental development builds layers of an application with a focus on the transport of data and cooperation, thereby enabling the management of systems complexity. It is quite common that this leads to a multi-tier architecture and object versioning. Typical incremental constructions appear in areas such as facility management [4]. A special *layer constructor* is widely used in frameworks, e.g. the OSI framework for communicating processes.

As an example consider the schema displayed in Figure 6 dealing with the handling of legal documents in the e-governance application SeSAM. It uses a specific composition frame, i.e. the type `DocumentInReviewing` is based on the type `ProposedDocument`. Legal documents typically employ particular document patterns, which are represented by the type `DocumentForm`. Actors in this applications are of type `Party`, which generalises `Person` and `Organisation`.

Loop Meta-Structures. Loop meta-structures appear whenever the lifespan of objects contains cycles. They are used for the representation of objects that store chains of events, people, devices, products, etc. Similar to the circulation meta-structure since it employs non-directional, non-hierarchical associations with different modes of connectivity being applicable. In this way temporal assignment and sharing of resources, association and integration, rights and responsibilities can be neatly represented and scaled.

Network Meta-Structures. Network or web meta-structures enable the collection of a network of associated types, and the creation of a multi-point web of associated types with specific control and data association strategies. The web has a specific data update mechanism, a specific data routing mechanism, and a number of communities of users building their views on the web.

As networks evolve quickly and irregularly, i.e. they grow fast and then are rebuilt and renewed, a network meta-structure must take care of a large number of variations to enable growth control and change management. Usually, they are supported by a multi-point center of connections, controlled routing and replication, change protocols, controlled assignment and transfer, scoping and localisation abstraction, and trader architectures. Furthermore, export/import converters and wrappers are supported. The database farm architecture [16] with check-in and check-out facilities supports flexible network extension.

3.3 Context Meta-structures

According to [18] we distinguish between the *intext* and the *context* of things that are represented as objects. Intext reflects the internal structuring, associations among types and subschemata, the storage structuring, and the representation options. Context reflects general characterisations, categorisation, utilisation, and general descriptions such as quality. Therefore, we distinguish between *meta-characterisation meta-structures* that are usually orthogonal to the intext structuring and can be added to each of the intext types, *utilisation-recording meta-structures* that are used to trace the running, resetting and reasoning of the database engine, and *quality meta-structures* that permit to reason on the quality of the data provided and to apply summarisation and aggregation functions in a form that is consistent with the quality of the data. The dimensionality of a schema permits the extraction of other context meta-structures [3].

Meta-Characterisation Meta-Structures. Meta-characterisation is orthogonal to the structuring dimension that may have led to a schema as displayed in Figure 1. They may refer to insertion/update/deletion time, keyword characterisation, utilisation pattern, format descriptions, utilisation restrictions and rights such as copyright and costs, and technical restrictions.

Meta-characterisations apply to a large number of types and should therefore be factored out. For instance, in an e-learning application learning objects, elements and scenes are commonly characterised by educational information such as interactivity type, learning resource type, interactivity level, age restrictions, semantic density, intended end user role, context, difficulty, utilisation interval restrictions, and pedagogical and didactical parameters.

Utilisation-Recording Meta-Structures. Logging, usage and history information is commonly used for recording the lifespan of the database. Therefore, we can distinguish between *history meta-structures* that are used for storing and recording the computation history within a small time slice, *usage-scene meta-structures* that are used to associate data to their use in a business process at a certain stage, a workflow step, or a scene in an application story, and record the actual usage.

Such meta-structures are related to one or more aspects of time, e.g. transaction time, user-defined time, validity time, or availability time, and associated with concepts such as temporal data types (instants, intervals, periods), and temporal statements such as current (now), sequenced (at each instant of time) and nonsequenced (ignoring time).

Quality Meta-Structures. Data quality is modelled by a variety of meta-structures capturing the sources (data source, responsible user, business process, source restrictions, etc.), intrinsic quality parameters (accuracy, objectivity, trustability, reputation, etc.), accessibility and security, contextual quality (relevance, value, timeliness, completeness, amount of information, etc.), and representation quality (ambiguity, ease of understanding, concise representation,

consistent representation, ease of manipulation). Data quality is essential whenever versions of data have to be distinguished according to their quality and reliability.

4 Application of Meta-structures in Data Modelling

Let us now briefly illustrate meta-structuring for some application examples.

4.1 Design by Units

The design-by-units framework [13] provides a modular design technique exploiting the internal skeleton structure of very large schemata. For illustration let us consider the example of the *Cottbusnet* website involving four main components:

- The star subschema characterising people maintains the data for types of people of interest: **Member_Of_Group**, **Representative_Of_Partner**, and **User**.
- The snowflake subschema on project information is used for the representation of information on various projects, their different stages, their results and their representation.
- The snowflake subschema on group information allows to store data on groups, their issues, leadership, obligations and results.
- The snowflake subschema on website maintenance provides data on the information that must be given through the web interface to authorized, anonymous and general users.

The skeleton of the application schema combines these components. The internal structure of the components is either a star or a snowflake subschema. The skeleton is associates the components using connector types, e.g. **Contribution** in Figure 2 between *Person* or *Member*, *Document*, *Project* and *Time*, **PortfolioProfile** between *Person*, *Group*, and *Website*, etc. In a similar way we can extent the skeleton by a component describing the organisation of group work.

4.2 Incremental Structuring

Meta-structuring supports the incremental evolution of database systems as a specific form of database system evolution, in particular for facility management systems. Such systems use a number of phases such as *planning phase*, *construction phase*, *realization phase*, and *maintenance phase*. Based on meta-structures we developed the novel architecture shown in Figure 7, which has already been positively evaluated in a project [8], in which auxiliary databases provide help information, and information on regulations, customers, suppliers, etc.

Thus, incremental evolution is supported by meta-structuring on the basis of import forms of two kinds:

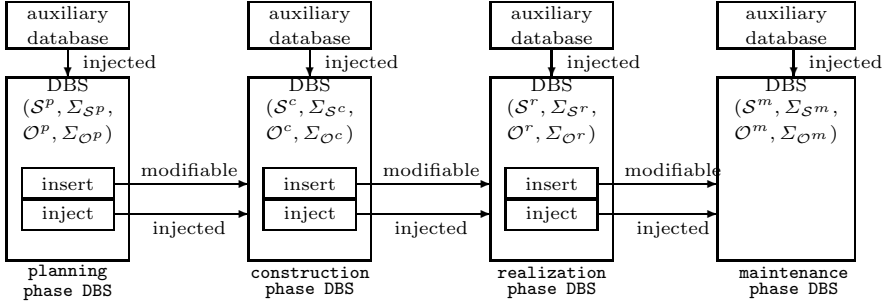


Fig. 7. The General Architecture of Incremental Evolution of Database Systems

- *Injection forms* enable the injection of data into another database. The forms are supported by cooperating views. Data injected into another database cannot be changed by the importing database system. The structure (S^{inject}, Σ_S) of the views of the exporting database system is entirely embedded into the structure $(S', \Sigma_{S'})$ of the importing database system. The functionality (O^{inject}, Σ_O) of the views of the exporting database system is partially embedded into the functionality $(O', \Sigma_{O'})$ of the importing database system by removing all modification operations on the injected data. These data can only be used for retrieval purposes.
- *Insertion forms* enable the insertion of data from an exporting database into an importing database. These data can be modified. The structure (S^{insert}, Σ_S) and the functionality (O^{insert}, Σ_O) of the views of the exporting database system are entirely embedded into the structure $(S', \Sigma_{S'})$ and the functionality $(O', \Sigma_{O'})$ of the importing database system.

4.3 The String Bag Modelling Approach

Looking from the distance at an ER schema the observer may find the diagram similar to a string bag, and indeed, there is a large number of similarities. Using the metaphor of a *string-bag* it has been observed that most database queries access types that are connected by a subbag [14], and this can be exploited to automatically derive corresponding SQL queries. For this some of the types in a query serve as critical types, while the others are used to select the appropriate paths in the ER schema.

This observation on query behavior can be generalized to database modelling declaring some of the types to be major or core types, while the others serve for specialising the application area. This forms the basis of an abstraction principle according to which the main view schemata form the abstraction of the schema, and the application is specified by the main views of the “handles”, i.e. the core types.

If we consider an application that addresses the types `Contact_Address`, `Party_Address`, `Geographical_Address`, then associations are view types on

the schema relating the main type **Address** to the kind of usage, i.e. to the main use of addresses in the application.

4.4 Rigid Structuring and Principles of Schema Abstraction

Database design techniques have often aimed at finding the best possible integration on the basis of the assumption of uniformity. As already observed database schemata should be based on a balance of three principles:

Autonomy: Objects represent things that are used in a separated and potentially independent form.

Hierarchy: The main classification method is the development of hierarchies. Hierarchies may be based on generalisation/specialisation, ontologies, or concept maps.

Coordination: Objects are related to each other and are used to exchange information among objects. The cohesion of objects is supported by coordination.

These principles can be extended to principles of schema abstraction:

Extraction of real differences: Recognition of differences enable the handling of classes that are to be treated differently. It strongly promotes eventual cohesion. Differences permit building the skeleton of the application.

Cultivation of hierarchies: Things in real applications can be associated to each other by specialisation and generalisation. Modelling uses hierarchies for factoring out facets and to use them for simpler and more efficient treatment.

Contraction by similarities: Unnecessary differentiation should be avoided. Therefore, similar subschemata and types should be clustered and modelled as components.

5 Conclusion

Very large database schemata with hundreds or thousands of types are usually developed over years, and then require sophisticated skills to read and comprehend them. However, lots of similarities, repetitions, and similar structuring elements appear in such schemata. In this paper we highlighted the frequently occurring meta-structures in such schemata, and classified them according to structure, lifespan and context. We demonstrated that meta-structures can be exploited to modularise schemata, which would ease querying, searching, reconfiguration, maintenance, integration and extension. Also reengineering and reuse are enabled.

In this way data modelling using meta-structures enables systematic schema development, extension and implementation, and thus contributes to overcome the maintenance problems arising in practice from very large schemata. Furthermore, the use of meta-structures also enables component-based schema development, in which schemata are developed step-by-step on the basis of the skeleton

of the meta-structure, and thus contributes to the development of industrial-scale database applications. We plan to elaborate further on formal aspects of meta-structing in data modelling with the concurrent submission in [6] being a first step. This will exploit graph grammars [2] and graph rewriting [12].

References

1. Brown, L.: *Integration Models – Templates for Business Transformation*. SAMS Publishing (2000)
2. Ehrig, H., Engels, G., Kreowski, H.-J., Rozenberg, G. (eds.): *Handbook of Graph Grammars and Computing by Graph Transformations. Applications, Languages and Tools*, vol. 2. World Scientific, Singapore (1999)
3. Feyer, T., Thalheim, B.: Many-dimensional schema modeling. In: Manolopoulos, Y., Návrát, P. (eds.) *ADBIS 2002*. LNCS, vol. 2435, pp. 305–318. Springer, Heidelberg (2002)
4. Kahlen, H.: *Integrales Facility Management – Management des ganzheitlichen Bauens*. Werner Verlag (1999)
5. Lenz, H.-J., Thalheim, B.: OLAP schemata for correct applications. In: Draheim, D., Weber, G. (eds.) *TEAA 2005*. LNCS, vol. 3888, pp. 99–113. Springer, Heidelberg (2006)
6. Ma, H., Schewe, K.-D., Thalheim, B.: *Handling meta-structures in data modelling* (submitted, 2008)
7. Moody, D.: *Dealing with Complexity: A Practical Method for Representing Large Entity-Relationship Models*. Ph.D thesis, University of Melbourne (2001)
8. Raak, T.: *Database systems architecture for facility management systems*. Master’s thesis, Fachhochschule Lausitz (2002)
9. Schewe, K.-D., Thalheim, B.: Component-driven engineering of database applications. In: *Conceptual Modelling – Proc. APCCM 2006*. CRPIT, vol. 53, pp. 105–114. Australian Computer Society (2006)
10. Schmidt, J.W., Sehring, H.-W.: Dockets: A model for adding value to content. In: Akoka, J., Bouzeghoub, M., Comyn-Wattiau, I., Métais, E. (eds.) *ER 1999*. LNCS, vol. 1728, pp. 248–263. Springer, Heidelberg (1999)
11. Siedersleben, J.: *Moderne Softwarearchitektur*. dpunkt-Verlag (2004)
12. Sleep, M.R., Plasmeijer, M.J., van Eekelen, M.C.J.D. (eds.): *Term Graph Rewriting – Theory and Practice*. John Wiley and Sons, Chichester (1993)
13. Thalheim, B.: *Entity Relationship Modeling – Foundations of Database Technology*. Springer, Heidelberg (2000)
14. Thalheim, B.: Generating database queries for web natural language requests using schema information and database content. In: *Applications of Natural Language to Information Systems – NLDB 2001*. LNI, vol. 3, pp. 205–209. GI (2001)
15. Thalheim, B.: Component construction of database schemes. In: Spaccapietra, S., March, S.T., Kambayashi, Y. (eds.) *ER 2002*. LNCS, vol. 2503, pp. 20–34. Springer, Heidelberg (2002)
16. Thalheim, B.: Component development and construction for database design. *Data and Knowledge Engineering* 54, 77–95 (2005)
17. Thalheim, B.: Engineering database component ware. In: Draheim, D., Weber, G. (eds.) *TEAA 2006*. LNCS, vol. 4473, pp. 1–15. Springer, Heidelberg (2007)
18. Wisse, P.: *Metapattern – Context and Time in Information Models*. Addison-Wesley, Reading (2001)

SOM-Based Dynamic Image Segmentation for Sign Language Training Simulator

Oles Hodych¹, Kostiantyn Hushchyn¹, Yuri Shcherbyna¹, Iouri Nikolski²,
and Volodymyr Pasichnyk²

¹ Ivan Franko Lviv National University, Ukraine
oles.hodych@gmail.com

² National University "Lvivska Politechnica", Ukraine
y_nikol@yahoo.com

Summary. The paper discusses an image segmentation algorithm based on Self-Organising Maps and its application for the improvement of hand recognition in a video sequence. The presented results were obtained as part of a larger project, which has an objective to build a training simulator for Ukrainian Sign Language. A particular emphasis in this research is made on the image preparation for Self-Organising Map training process for the purpose of successful recognition of image segments.

Keywords: image segmentation, self-organising maps.

1 Introduction

Sign languages are based on hand signs, lip patterns and body language instead of sounds to convey meaning. The development of sign languages is generally associated with deaf communities, which may include hearing or speech impaired individuals, their families and interpreters. The only currently viable ways to enable communication between hearing impaired and not impaired people is to use services provided by interpreters (specially trained individuals or software applications), or to learn a sign language.

A manual communication has developed in situations where speech is not practical. For instance, scuba diving or loud work places such as stock exchange. In such cases learning a sign language is the most effective solution.

There is a number of commercial software packages and research projects directed at developing software applications for sign language interpretation. The majority of such software is directed at interpreting a spoken to sign language, which covers all major cases where interpretation is required for deaf people (e.g. conventions, television). For example, researchers at IBM have developed a prototype system, called SiSi (say it, sign it), which offers some novel approaches by utilising avatars to communicate speech into sign language [16]. Some other applications, such as iCommunicator [21], are based on a database of sign language videos and provide means to adaptively adjust to user's speech.

Authors of this paper are conducting a research for the purpose of creating an adaptive sign language training simulator. A large part of the core ideas and technologies

produced is already discussed in a number of articles as well as demonstrated at CeBIT 2006, 2007 and 2008. The main motivation behind this research is to provide an affordable solution for people to use the end product for self-training of the Ukrainian sign language.

One of the core ideas of this project is to use a motion camera (such as a web camera) for capturing user's hand gestures, recognise them and match against existing samples for providing trainee with a feedback as to how well the gesture was performed. When successfully implemented this approach should provide users with a self-training easy to set up environment requiring no coaching by a human trainer.

The detailed discussion of the project's progress and its key results are covered in several articles [3] [4] [5]. Gesture is a typical element of a sign language. One of the most difficult tasks, which were encountered during the research, was the requirement to recognise a sign from a sequence of video frames regardless of the background on which the gesture was performed. Our tests reveal that the adaptive technique employed for recognising user's hand (and thus the gesture) would provide a much higher success rate when performing recognition on frames with the background used for its training. The proposed training simulator should cater for a wide range of situations including different backgrounds. For example, the user of the system might be wearing a stripy shirt one day and a plain white t-shirt some other time. Thus, there was formulated a requirement to filter out the background in frames in a video sequence before attempting to recognise the gesture. During the research we've come to realize that instead of trying to identify what is the background, it is potentially more efficient to identify the smallest possible area on the image (frame), which contains the hand. The identified area could then be used for further processing to actually identify the gesture using one of the proposed earlier methods utilising the dactyl matching [5]. The process of partitioning a digital image into multiple regions is known as image segmentation. Image segmentation is typically used for locating objects or boundaries in the image, which is what's needed to locate a hand in our case.

2 Related Work

There are several ways to approach the problem of image segmentation. One possible way, which is more inline with our research, is to treat image segmentation as a clustering task, where objects for grouping are image pixels, and the actual groups (or clusters) are image segments. Hence the term *image clustering*, which refers to means for high-level description of the image content. Self-Organising Map and its variations were the subject of our research for the past several years [10] [11] [12], and therefore it was a natural choice for the task of image clustering.

Image segmentation is a popular research subject and there is a large number of related research projects as well as readily available commercial and open source tools. In this section we present a short overview of the published research results dedicated specifically to the use of Self-Organising Maps (SOM) as a technology for image clustering.

In [17] authors proposed the use of the two-stage process based on SOM with one-dimensional lattice. The SOM network is trained during the first stage, and in the second stage it is clustered using K-means algorithm in order to determine the number of image segments. One of the main disadvantages of the proposed technique is the use of the reduced colour information where only hue and saturation were used for preparing the data source (luminance component was excluded). As discussed later, the colour space plays a significant role in preparing the data for image clustering.

In [6] and [7] researchers present a unique data preparation scheme where not only the colour, but also the image texture was used, which proved to yield a success rate of 61.3% (with texture) comparing to 53.6% while using only colour information. Provided tests included very complex outdoor images.

A large number of human image processing approaches utilise skin detection as a key element for feature extraction. The histograms and Gaussian mixture models are amongst the most popular colour modeling methods. However, these techniques are not always best suited in the real life dynamic environments. In [2] authors proposed a SOM-based algorithm for skin detection. The accuracy of 94% was claimed on facial images.

In [1] a multi-stage clustering algorithm was proposed in application to colour image segmentation. The first stage of the proposed algorithm utilises SOM due to its distinct features of reducing the sample size and at the same time preserve the input distribution. The subsequent stages could utilise segmentation techniques that would not be possible on samples of a large size. The RGB colour space was used in this research.

When processing an image for a segmentation task it should be presented in a suitable for the segmentation algorithm way. Very often a three dimensional RGB space is used, where each or group of pixels on the image is represented as a three dimension vector. In [18] authors used both colour and spatial information. Thus utilising five dimensional vectors (X, Y, R, G, B) for representing image data used in SOM training. In addition, a merging algorithm was introduced for clustered blocks to be combined into a specified number of regions with some semantic means.

Authors of [20] concentrated their research on the use of one-dimension SOM network for image segmentation. According to this research the best results were obtained for SOM with lattice configuration where the first and the last neurons are linked forming a circular structure.

Paper [22] discusses the use of an adaptive SOM colour segmentation algorithm. Similarly to [20] one dimensional SOM was used, but in this case it supported growing – pruning and merging facilities were developed to find an appropriate number of clusters automatically. Additionally, in order to further improve results in light and background changing conditions, authors developed, what they call, a *transductive* algorithm to learn the dynamic colour distribution in HSI colour space by combining supervised and unsupervised learning paradigms. The presented results yield an excellent performance. However, most of the presented tests utilised images either of a small size, thus automatically reducing the complexity, or those with a close to uniform background.

The main objective of the proposed in this paper approach is to identify a segment containing human hand in a video sequence with a non-uniform background in

a timely from the processing speed perspective manner. The following sections discuss the three key aspects of the proposed approach – selection of a colour space for image representation, data reduction for SOM training, and an information generalisation based on one video frame for segmentation of a complete sequence.

3 Dynamic Image Clustering

The discussion of the theoretical and algorithmic foundation of the conducted research is presented in two sections. The first sections addresses the data preparation. Specifically, the development of the most adequate method for transforming images from a video sequence into a vector space suitable for SOM consumption (i.e. training and interpretation). This phase directly effects the quality of image clustering.

3.1 Data Preparation

The first stage of data preparation is to choose a vector space for representing each pixel on the image. The processing speed is of high importance for fulfilling the requirement of real-time processing. The training of SOM is the most time consuming operation. Therefore, the second stage of data preparation addresses the reduction of the number of data samples used for training.

Colour space. It is well known that SOM operates based on the principles of the brain. One of the key SOM features is preservation of the topological order of the input space during the learning process. Some relatively recent research of the human brain has revealed that the response signals are obtained in the same topological order on the cortex in which they were received at the sensory organs [19]. One of such sensory organs are eyes, thus making the choice of SOM for analysing the visual information one of the most natural.

Colour is the brain's reaction to specific visual stimulus. Therefore, in order to train SOM for it to reflect the topological order of the image perceived by a human eye, it is necessary to choose the colour space, which closely models the way sensors obtain the visual information. The eye's retina samples colours using only three broad bands, which roughly correspond to red, green and blue light [8]. These signals are combined by the brain providing several different colour sensations, which are defined by the CIE (Commission Internationale de l'Eclairage (French), International Commission on Illumination) [15]: Brightness, Hue and Colourfulness. The CIE commission defined a system, which classifies colour according to the human visual system, forming the tri-chromatic theory describing the way red, green and blue lights can match any visible colour based on the eye's use of three colour sensors.

The colour space is the method, which defines how colour can be specified, created and visualised. As can be deduced from the above, most colour spaces are three-dimensional. There are more than one colour space, some of which are more suitable for certain applications than others. Some colour spaces are perceptually linear, which means that an n -unit change in stimulus results in the same change in perception no

matter where in the space this change is applied [8]. The feature of linear perception allows the colour space to closely model the human visual system. Unfortunately, the most popular colour spaces currently used in image formats are perceptually nonlinear. For example, BMP and PNG utilise RGB¹ colour space, JPEG utilises YCbCr, which is a transformation from RGB, HSL² is another popular space, which is also based on RGB.

The CIE based colour spaces, such as CIEluv and CIELab, are nearly perceptually linear [8], and thus are more suitable for the use with SOM. The CIEXYZ space devises a device-independent colour space, where each visible colour has nonnegative coordinates X, Y and Z [14]. The CIELab is a nonlinear transformation of XYZ onto coordinates L^*, a^*, b^* [14].

The image format used in our research is uncompressed 24-bit BMP (8 bit per channel), which utilises the RGB colour space. In order to convert vectors $(r, g, b) \in RGB$ into $(L^*, a^*, b^*) \in CIELab$ it is necessary to follow an intermediate transformation via the CIE XYZ colour space. These transformations are described in details in [13] and [14]. Application of the two-step transformation to each pixel of the original image in RGB space produces a transformed image in CIELab space used for further processing.

It is important to note that when using SOM it is common to utilise Euclidean metric for calculation of distances during the learning process [19]³. Conveniently, in CIELab space the colour difference is defined as Euclidean distance [14].

In order to demonstrate significance of the colour space selection please consider images depicted on Fig. 1.

The original image was 800×600 (refer Fig. 1(a)). Three datasets were composed based on this image using three different colour spaces: RGB, HSL and CIELab. SOM then was used to cluster these datasets. The result of this clustering is depicted on Fig. 1(b) for RGB-based dataset, Fig. 1(c) for HSL-based dataset, and Fig. 1(d) for CIELab-based dataset. As can be easily observed the CIELab-based image representation provides the best result.

Training dataset composition. Instead of using every image pixel for the SOM training process, the following approach was employed to reduce the number of data samples in the training dataset.

The basic idea is to split an image into equal segments $n \times n$ pixels. Then for each such segment find two the most diverged pixels and add them to the training dataset. Finding the two most diverged pixels is done in terms of the distance applicable to the colour space used for image representation. Due to the fact that each pixel is a three dimensional vector, each segment is a matrix of vector values. For example, below is an image A of 4×4 pixels in size represented in the CIELab space, and split into four segments 2×2 pixels each.

¹ Uncompressed BMP files, and many other bitmap file formats, utilise a colour depth of 1, 4, 8, 16, 24, or 32 bits for storing image pixels.

² Alternative names include HSI, HSV, HCL, HVC, TSD etc. [8]

³ The selection of the distance formula depends on the properties of the input space, and the use of Euclidean metric is not mandatory.

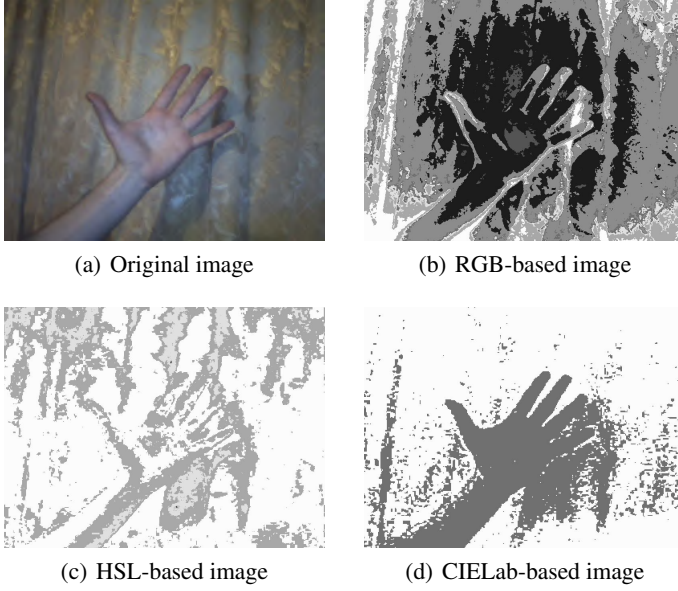


Fig. 1. Image SOM processing for different colour spaces

$$A = \begin{pmatrix} (L_1^1, a_1^1, b_1^1)^T & (L_2^1, a_2^1, b_2^1)^T & (L_3^1, a_3^1, b_3^1)^T & (L_4^1, a_4^1, b_4^1)^T \\ (L_1^2, a_1^2, b_1^2)^T & (L_2^2, a_2^2, b_2^2)^T & (L_3^2, a_3^2, b_3^2)^T & (L_4^2, a_4^2, b_4^2)^T \\ (L_1^3, a_1^3, b_1^3)^T & (L_2^3, a_2^3, b_2^3)^T & (L_3^3, a_3^3, b_3^3)^T & (L_4^3, a_4^3, b_4^3)^T \\ (L_1^4, a_1^4, b_1^4)^T & (L_2^4, a_2^4, b_2^4)^T & (L_3^4, a_3^4, b_3^4)^T & (L_4^4, a_4^4, b_4^4)^T \end{pmatrix}$$

Thus, the first segment is:

$$S_1 = \begin{pmatrix} (L_1^1, a_1^1, b_1^1)^T & (L_2^1, a_2^1, b_2^1)^T \\ (L_1^2, a_1^2, b_1^2)^T & (L_2^2, a_2^2, b_2^2)^T \end{pmatrix}$$

The above approach can be summarised as the following algorithm. Let n denote the size of segments used for image splitting, the value of which is assigned based on the image size. T – the training set, which is populated with data by the algorithm. Let's also denote j th pixel in segment S_i as $S_i(j)$. Further in the text both terms *pixel* and *vector* are used interchangeably.

The above algorithm provides a way to reduce the training dataset. It is important to note that an excessive reduction could cause omission of significant pixels resulting in poor training. At this stage it is difficult to state what rule can be used to deduce the optimal segment size. The segmentation used for the presented results was obtained though experimentation. However, even applying segmentation 2×2 pixels to an image of 800×600 pixels in size reduces the training dataset from 460000 down to 240000

Algorithm 1. Training dataset composition

Initialisation. Split image into segments of $n \times n$ pixels; $N > 0$ – number of segments; $T \leftarrow \emptyset$; $i \leftarrow 1$.

1. Find two the most diverged pixels $p' \in S_i$ and $p'' \in S_i$ using Euclidian distance.
 - 1.1 $max \leftarrow -\infty$, $j \leftarrow 1$
 - 1.2 $k \leftarrow j + 1$
 - 1.3 Calculate distance between pixels $S_i(j)$ and $S_i(k)$: $dist \leftarrow \|S_i(j) - S_i(k)\|$
 - 1.4 If $dist > max$ then $p' \leftarrow S_i(j)$, $p'' \leftarrow S_i(k)$ and $max \leftarrow dist$
 - 1.5 If $k < n \times n$ then $k \leftarrow k + 1$ and return to step 1.3
 - 1.6 If $j < n \times n - 1$ then $j \leftarrow j + 1$ and return to step 1.2
 2. Add $p' \in S_i$ and $p'' \in S_i$ to the training set: $T \leftarrow T \cup \{p', p''\}$
 3. Move to the next segment $i \leftarrow i + 1$. If $i \leq N$ then return to step 1, otherwise stop.
-

elements, which in turn enables the use of a smaller lattice and reduces the processing time required for SOM training.

3.2 Interpretation of Clusters

There are several aspects to a successful application of SOM, among which are:

- Self-organisation process, which encompasses a problem of selecting a learning rate and a neighbourhood function.
- The size and structure of the SOM lattice.

In this research the guidelines from [19] and [11] were followed to conduct the self-organisation process. The structure of the SOM lattice may differ in its dimensionality and neighbourhood relation between neurons. The use of 2-dimensional lattice with hexagonal neighbourhood relation proved to be the most efficient in our research producing more adequate clustering results comparing to other evaluated configurations.

Once the SOM structure and parameters for self-organisation process are selected, the SOM is trained on the training set T , which is composed for the image to be clustered. The trained SOM is then used for the actual image clustering.

As has been mentioned in previous sections, one of the most important features of SOM is topology preservation. This feature is fundamental to the proposed image segmentation approach. The basic underlying principles of which are:

- Image pixels represented by topologically close neurons should belong to the same cluster and therefore segment.
- The colour or marker used for segment representation is irrelevant as long as each segment is associated with a different one.

These two principles suggest that the position of neurons in the lattice (i.e. coordinates on the 2D plane) can be used for assigning a marker to a segment represented by any particular neuron instead of the neurons' weight vectors. This way weight vectors are used purely as references from 2D lattice space into 3D colour space, and neural

locations represent the image colour distribution. As the result of a series of conducted experiments the following formulae for calculating an RGB colour marker for each neuron have produced good results.

$$R_j \leftarrow x_j + y_j \times \lambda; G_j \leftarrow x_j + y_j \times \lambda; B_j \leftarrow x_j + y_j \times \lambda; \quad (1)$$

In formula (1) values x_j and y_j are coordinates of neuron $j = \overline{1, M}$, where M is the number of neurons in SOM. Constant λ should be greater or equal to the diagonal of the SOM lattice. For example, if SOM lattice has a rectangular shape of 16×16 neurons then λ could be set to 16. Applying the same formula for R, G, and B components produces a set of gray scale colours. However, each neuron has its own colour, and one of the currently not fully resolved issues is how to group neurons based on the assigned colours into larger segments. There are several approaches, which are being currently developed to provide automatic grouping of SOM neurons into clusters [12]. However, the presented in this paper results were obtained by applying a threshold to the segmented with SOM image, which requires human interaction in specifying the threshold value. The presented image segmentation approach can be summarised as the following algorithm.

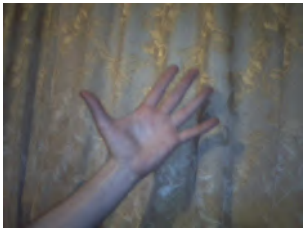
Algorithm 2. Image segmentation

Initialisation. $p_j = (R_j, G_j, B_j)$ – pixel j ; $j = \overline{1, K}$; $K > 0$ – total number of pixels; $j \leftarrow 1$; $i^*(p_j) = (R_{i^*}, G_{i^*}, B_{i^*})$ – a weight vector of the best matching unit (BMU – winning neuron) for input vector p_j ; (x_{i^*}, y_{i^*}) – coordinates of neuron i^* ; choose appropriate values for λ .

1. Find $BMU(p_j)$ for vector p_j in the trained SOM utilising the distance used for training (Euclidian for CIELab).
 2. Calculate marker for pixel p_j : $R_j \leftarrow x_{i^*} + y_{i^*} \times \lambda$, $G_j \leftarrow R_j$, $B_j \leftarrow R_j$.
 3. Move to the next image pixel: $j \leftarrow j + 1$;
 4. If $j \leq K$ return to step 1, otherwise stop.
-

4 Experimental Results

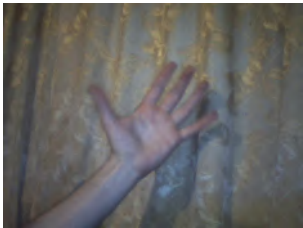
In this section we would like to demonstrate some results of this research. A special interest is the case of training SOM on one of the frames in a video sequence and using it for segmentation of subsequent frames. The following figures present results by depicting the original and segmented images, which correspond to frames number 25 through to 60 with step of 5 frames of the recorded video. The training of SOM and determining of the appropriate threshold value was performed only on the first image (i.e. 25th frame). The recorded video captured an open palm closing and opening again during a period of several seconds. The recording was done using an ordinary PC web camera capable of 30FPS throughput with a frame size of 800×600 pixels. The background of the captured scene is nonuniform, which increases the complexity of image segmentation.



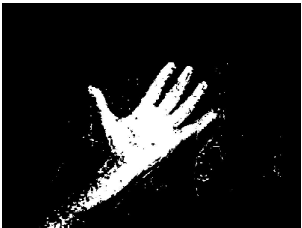
(a) Orig. frame 25



(b) Segmented frame 25

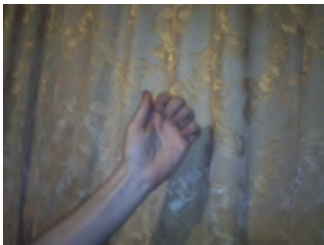


(c) Orig. frame 30

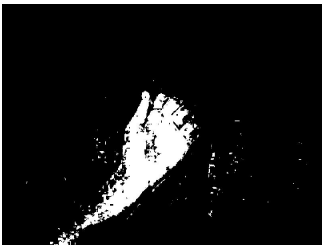


(d) Segmented frame 30

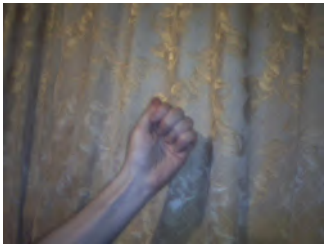
Fig. 2. Frame 25 and 30



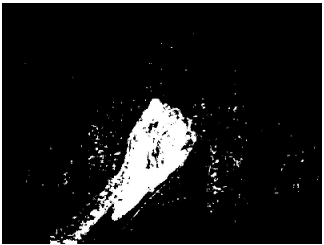
(a) Orig. frame 35



(b) Segmented frame 35

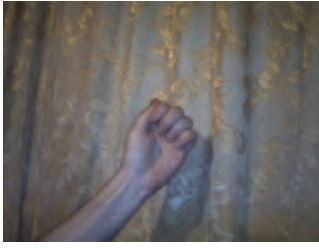


(c) Orig. frame 40



(d) Segmented frame 40

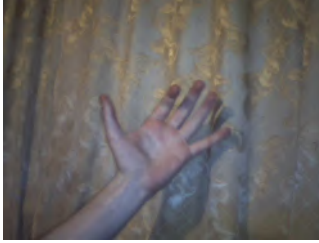
Fig. 3. Frame 35 and 40



(a) Orig. frame 45



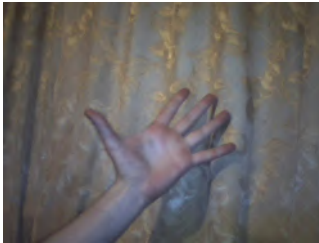
(b) Segmented frame 45



(c) Orig. frame 50



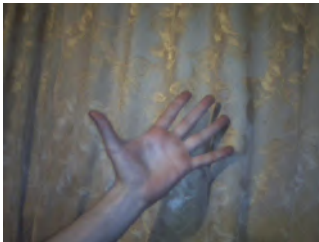
(d) Segmented frame 50

Fig. 4. Frame 45 and 50

(a) Orig. frame 55



(b) Segmented frame 55



(c) Orig. frame 60



(d) Segmented frame 60

Fig. 5. Frame 55 and 60

Figure 2 depicts a fully open palm corresponding to frame 25, which was used for SOM training, and frame 30 depicting a palm with slightly contacted fingers.

Frames 35 and 40 correspond to palm closing, which are depicted in figures 3. As can be observed, the contracted palm is successfully separated from the background by the proposed segmentation algorithm. However, at the same time a slightly greater number of artifacts, which do not belong to the palm, are captured.

Figures 4 depict frames 45 and 50 where palm started opening again.

The remaining figure 5 depicts the final frames that captured the full opening of the palm. These frames are very similar to frames 25 and 30, therefore, the good segmentation results were expected.

The key aspect of the presented in this section results is the use of SOM trained only on a single frame. This initial frame as well as all subsequent ones have been successfully segmented with clear separation of the human palm from the nonuniform background. Although, some elements of the background were recognised as part of the same segment and caused minor undesired artifacts scattered around the palm. The use of only one frame for SOM training provides a provision for much faster dynamic image segmentation needed for video, avoiding SOM retraining for every frame.

5 Conclusion and Future Work

The main purpose of developing an image segmentation algorithm in our case is to improve image analysis for dactyl matching. The proposed approach showed good results not only for human hand recognition, and potentially can be used for other applications. The main disadvantage of the developed approach is the need for human interaction when specifying the threshold values in the final step of image segmentation. However, this aspect is being currently addressed utilising the results obtain in [12], which allows automatic clustering of the trained SOM. Another important subject of the future research direction is increasing the quality of segmentation by applying hierarchical clustering. The basic idea behind this approach is to start SOM training on images with reduced information, following additional training based on the same image with increased information. There are many image smoothing methods that provide a way of controlling the amount of image details, which may impact the quality of information reduction and thus segmentation results.

References

1. Akgül, C.B.: Cascaded self-organizing networks for color image segmentation (2004), http://www.tsi.enst.fr/~akgul/oldprojects/CascadedSOM_cba.pdf
2. Brown, D., Craw, I., Lewthwaite, J.: A SOM Based Approach to Skin Detection with Application in Real Time Systems, University of Aberdeen (2001), http://www.bmva.ac.uk/bmvc/2001/papers/33/accepted_33.pdf
3. Davydov, M.V., Nikolskyi, Y.V.: Automatic identification of sign language gestures by means on dactyl matching. Herald of National University "Lvivska Polytechnica" 589, 174–198 (2007)
4. Davydov, M.V., Nikolskyi, Y.V., Pasichnyk, V.V.: Software training simulator for sign language learning. Connection, 98–106 (2007) (in Ukrainian)

5. Davydov, M.V., Nikolskiy, Y.V., Pasichnyk, V.V.: Selection of an effective method for image processing based on dactyl matching for identification of sign language gestures. Herald of Kharkiv National University of Radio-Electronics 139, 59–68 (2008) (in Ukrainian)
6. Campbell, N.W., Thomas, B.T., Troscianko, T.: Neural Networks for the Segmentation of Outdoor Images. In: International Conference on Engineering Applications of Neural Networks, pp. 343–346 (1996)
7. Campbell, N.W., Thomas, B.T., Troscianko, T.: Segmentation of Natural Images Using Self-Organising Feature Maps, University of Bristol (1996)
8. Ford, A., Roberts, A.: Colour Space Conversions (1998),
<http://www.poynton.com/PDFs/coloureq.pdf>
9. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, San Francisco (2001)
10. Hodych, O., Nikolskiy, Y., Shcherbyna, Y.: Application of Self-Organising Maps in medical diagnostics. Herald of National University “Lvivska Polytechnica” 464, 31–43 (2002)
11. Hodych, O., et al.: Analysis and comparison of SOM-based training algorithms. Control Systems and Machines 2, 63–80 (2006) (in Ukrainian)
12. Hodych, O., et al.: High-dimensional data structure analysis using Self-Organising Maps. In: 9th International Conference, CAD Systems in Microelectronics. CADSM apos 2007, February 2007, pp. 218–221 (2007)
13. Hoffmann, G.: CIE Color Space (2000),
<http://www.fho-emden.de/~hoffmann/ciexyz29082000.pdf>
14. Hoffmann, G.: CIELab Color Space (2003),
<http://www.fho-emden.de/~hoffmann/cielab03022003.pdf>
15. Hunt, R.W.G.: Measuring Colour, 3rd edn. Fountain Pr Ltd. (2001)
16. IBM Research Demonstrates Innovative Speech to Sign Language Translation System, Press-release (September 12, 2007),
<http://www-03.ibm.com/press/us/en/pressrelease/22316.wss>
17. Moreira, J., Da Fontoura Costa, L.: Neural-based color image segmentation and classification using self-organizing maps (1996),
<http://mirror.impa.br/sibgrapi96/trabs/pdf/a19.pdf>
18. Jiang, Y., Chen, K.-J., Zhou, Z.-H.: SOM Based Image Segmentation. LNCS (LNAI), vol. 2639, pp. 640–643. Springer, Heidelberg (2003)
19. Kohonen, T.: Self-Organizing Maps, 3rd edn. Springer, Heidelberg (2001)
20. Reyes-Aldasoro, C.C.: Image Segmentation with Kohonen Neural Network Self-Organising Maps (2004),
<http://www.cs.jhu.edu/~cis/cista/446/papers/SegmentationWithSOM.pdf>
21. The iCommunicator User’s Guide (2005),
<http://www.mycommunicator.com/downloads/iCommunicator-UserGuide-v40.pdf>
22. Wu, Y., Liu, Q., Huang, T.S.: An Adaptive Self-Organizing Color Segmentation Algorithm with Application to Robust Real-time Human Hand Localization. In: Proc. Asian Conf. on Computer Vision, Taiwan, (2000)

Agile Software Solution Framework: An Analysis of Practitioners' Perspectives

Asif Qumer and Brian Henderson-Sellers

Faculty of Engineering and Information Technology, University of Technology, Sydney
Broadway, NSW 2007, Australia
{asif,brian}@it.uts.edu.au

Abstract. We have developed an agile software solution framework (ASSF) to create and tailor situation-specific agile methods by using a method engineering approach. Here, we report on a questionnaire-based survey with thirty-three experts in order to determine the relevance and importance of the aspects or elements of agile software development methodology specified in ASSF. We have analysed the relevance and importance that each respondent places on the identified elements of the ASSF.

Keywords: Agile Software Solution Framework, Agile Methods, Method Engineering.

1 Introduction

Agile software development methods have several benefits over traditional plan-based methods - in particular, their ability to handle projects where the project requirements are not fixed [9, 10, 16, 17]. While many organizations are interested in adopting agile methods for their software development projects, there is little guidance available on how to do so. In our current research, we have developed an Agile Software Solution Framework (ASSF: [13]), which provides both a conceptual underpinning and an overall context for the exploration of agile methods in practice (Figure 1). The main components of the ASSF are the Agile Conceptual Aspect Model together with links to tools (Agile Toolkit, 4-DAT) and a bridge to Business. The tools are described in [12] and [14] respectively. They support quantitative evaluation of the contributory elements of an agile approach – characteristics such as leanness, flexibility and responsiveness [10]. Within the conceptual model, the main elements are agile knowledge, agile governance and agile method core, in the context of a particular abstraction mechanism, such as objects, agents, services etc. This abstraction concept is then linked to an appropriate software technology.

A number of iterative assessment and validation experiments have been performed to explore and evaluate the construct and effectiveness of the ASSF; in order to validate the usability of the ASSF, we have applied it empirically to the construction of agile software development processes in industry, on several pilot projects [5]. We have illustrated [13] how two organizations were able to transform from a traditional development environment to a more agile one by using the ASSF. We used a people-oriented approach (involving software industry representatives who will be the users

of the framework) and a communication-oriented approach (to get feedback from industry representatives as well as researchers) to iteratively develop the components of the ASSF. In order to determine the relevance and importance of the elements of the ASSF, an administered questionnaire-based survey has been used iteratively to collect data from thirty-three experts, belonging to ten different countries. The main objective of this study was to involve the opinion of the experts both from the software industry and the research community in designing and testing the ASSF, these opinions helping us to assess the construct validity and external validity of our approach.

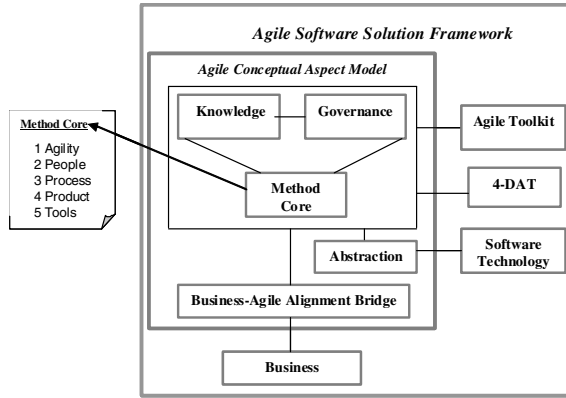


Fig. 1. The main element of the ASSF (after [13])

This paper presents the analysis and results of the part of this study that is focused on determining which agile software methodology aspects (agile or non-agile fragments) are most relevant and important to form/represent the agile methods from a practitioner's point of view. The findings of this study, combined with the earlier published results of the empirical evaluation of the ASSF [13], will facilitate method engineers in making a decision about the importance of the aspects (from more critical to less) or to which methodology aspects to choose for a particular situation when using a situational method engineering approach [3].

The following questions have been used to report the first part of this analysis.

- In your opinion, what are the main aspects of a software development methodology?
- What are the main factors that you will consider for the selection or construction of a software methodology for a specific project?
- How important is the role of a light-weight governance approach in an agile software development organization to bring in sufficient discipline and rationale to an agile methodology?
- How important is the use of a light-weight knowledge engineering and management approach in an agile software development organization for the purpose of learning and decision making?

- What do you think about the alignment of the business values and agile practices?
- How important is the impact of business policies, goals, strategies and culture in agile software development adoption, execution and governance?

This paper is organized as follows: Section 2 presents the research methodology. Section 3 presents the analysis and results of this study. Finally, in Section 4, we illustrate the validity and limitations of this study; before concluding in Section 5.

2 Research Methodology

A close-ended questionnaire has been iteratively used as a data collection instrument in this study. The questionnaire was designed to elicit the relevance and importance that each respondent places on each of the main identified elements of the ASSF (Figure 1). The design of the questions is based on the already published work in the field of software process. The questionnaire has been examined for the support of the objective of this research.

The response to the questions was gathered at different times in 2006 and 2007. A questionnaire-response repository has been created to store the responses of the re-search participants. Later, two statistical tools (SPSS and MS Excel) were used for the purpose of data analysis and representation.

The questionnaire was distributed at conferences as well as in research and soft-ware development organizations. In total, 33 people (from medium to large size or-ganizations) from ten different countries participated in this survey-based research. The data were collected from the participants who were involved in software process at different levels and could be the potential users of the ASSF.

Table 1. Profile of the participants

Countries	Response	Percentage
Australia	17	52%
UK	2	6%
Norway	3	9%
Finland	2	6%
Spain	1	3%
Crotia	1	3%
USA	1	3%
Sweden	3	9%
Turkey	2	6%
China	1	3%
Total	33	
Type	Response	Percentage
Research	18	55%
Industry	15	45%

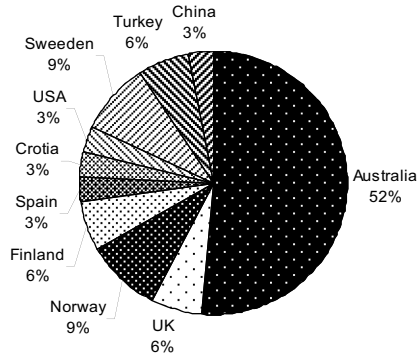


Fig. 2. Profile of the participants

In order to make the participant sample fairly representative and as un-biased as possible, different people from industry and research organizations were selected, which included agile consultants, coaches, managers, research scientists and developers. However, this research does not claim that this is a truly statistically representative sample as a fully representative sample is hard or impossible to get ([4]: research methods and statistics). This research has used the convenience sample rather than a random sample since a response was sought from a person with a specific role both in industry or research. The participants and their responses for the purpose of this research study can be grouped into two main categories. Table 1 and Figure 2 show the profile of the participants.

However, the participant responses were not sorted between researchers and industry professionals as the idea is not to compare the differences in perceptions and opinions across these two groups. In addition to that, responses were not segmented based upon organizational sizes, as organization size was anticipated as having little impact on the overall knowledge of the software process and their (research participants) personal opinions i.e. the opinion of an individual.

3 Analysis and Results

Thirty-three survey responses were analysed in order to get the opinion of the participants from these ten different countries. The response was purely based on their personal experiences and knowledge – although it should be acknowledged that there is always the possibility in this type of questionnaire survey that not all participants will understand the question in exactly the same way. Analysis of the response allowed us to identify and determine the relevance and importance of the elements of the ASSF. The following sections and sub-sections examine and analyse each question's response in detail.

3.1 Software Methodology Core Aspects

In this section, the response received by asking the first question has been analysed. The question is: In your opinion, what are the main aspects of a software development methodology?

Table 2 and Figure 3 present the identified list of the core methodology aspects, their relevance importance and the percentages. Here, it may be observed from these data that the aspect of highest importance (24 out of 33 responses, which is 73%) is the ‘process’ aspect, whereas the ‘tools’ aspect is considered the lowest (6 out of 33 responses which is only 18%) in the core aspects of a software development methodology. The ‘people’ aspect and the newly identified ‘agility’ aspect are almost at the same level of relevance and importance. It can be seen from this analysis and, although agile methods are people-focused, we cannot ignore the importance of the process aspect in a software development methodology. The ASSF has been designed to represent all these elements in a software development methodology with a different level of importance. The patterns of relative importance may be helpful to practitioners in optimizing their agile approach.

Table 2. Software methodology core aspects

Aspects	Importance	Percentage
Process	24	73%
Tools	6	18%
People	16	48%
Products	9	27%
Agility	15	45%
Abstraction	12	36%

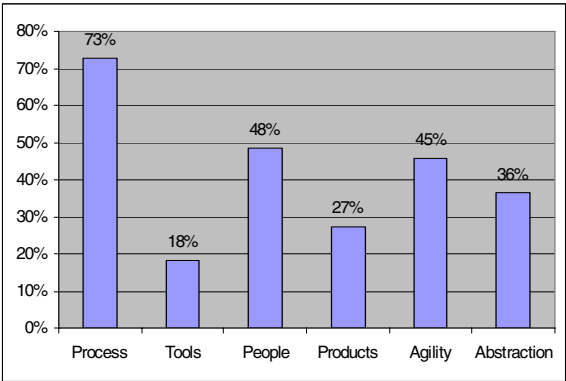


Fig. 3. Software methodology core aspects

3.2 Software Methodology Construction and Selection Factors

In this section, the response received on the second question has been analysed. The question is: What are the main factors that you will consider for the selection or construction of a software methodology for a specific project?

This question focuses on identifying the important factors that a method engineer or process engineer may consider while creating or selecting a methodology for a specific situation. However, although all the factors listed below are important, based on the response of the research participants and the collected data (Table 3 and Figure 4), it can be observed that project size and complexity are the most important factors and could significantly affect the design or selection of a methodology for a particular project or situation.

Table 3. Software methodology construction and selection factors

Factors	Importance	Percentage
Project size, complexity	30	91%
Degree of agility of a method	17	52%
Team size, people skills	18	55%
Tools, development environment	13	39%
Culture and physical environment	13	39%
Abstraction mechanism	8	24%
Constraints and risks	18	55%
Others	4	12%

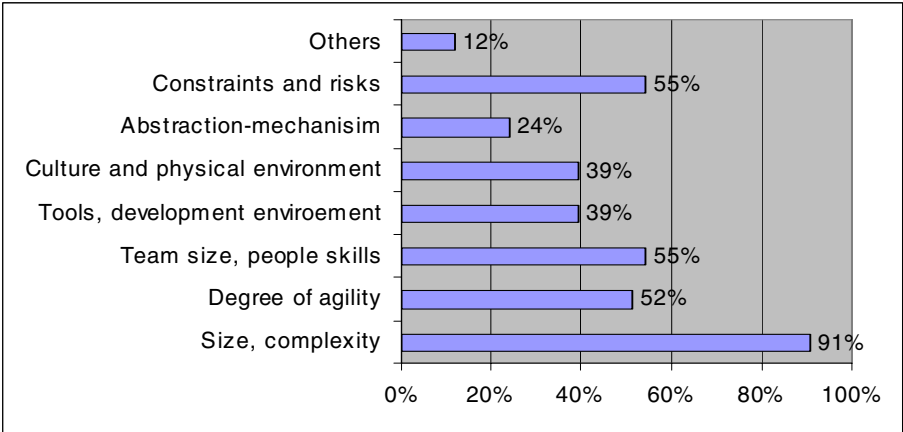


Fig. 4. Software methodology construction and selection factors

3.3 Agile Governance

In this section, the third question has been analysed. The question is: How important is the role of a light-weight governance approach in an agile software development organization to bring in sufficient discipline and rationale to an agile methodology?

The first two questions focus on the core aspects of a software development methodology and the methodology construction or selection factors that can play an important role in combining the methodology aspects to create a methodology. Here, in this question, the role of the governance in agile software development is determined to test the following hypothesis:

“Light-weight governance will bring sufficient control and rationale in agile methodology, which in turn will enable the agile methods to be scaled up for larger developments and projects”.

According to this survey response (Table 4 and Figure 5), 87% (48+33+ 6) of the participants reported and recognized the importance of governance at three different levels (very important, important and minimally important). This recognition is important for bringing in rationale and control into agile methods to scale them for large and complex projects. However, only 12% of the participants were neutral (neither important nor unimportant) and 0% of the participants disagreed. It can be seen from this analysis and data that it is important to include the governance factor in a software methodology together with the core aspects of a software methodology. This clearly justifies the inclusion of the ‘agile governance’ aspect in the ASSF (cf. [11]).

Table 4. Governance in agile methods

Governance	Response	Percentage
Very important	16	48%
Important	11	33%
Minimal important	2	6%
Neutral	4	12%
Not important	0	0%

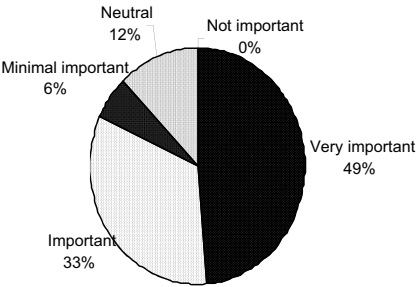


Fig. 5. Governance in agile methods

3.4 Agile Knowledge

In this section, the fourth question has been analysed. The question is: How important is the use of a light-weight knowledge engineering and management approach in an agile software development organization for the purpose of learning and decision making?

ASSF includes the aspect of ‘agile knowledge’; this question has been used to highlight the importance of the inclusion of knowledge engineering approach in constructing a knowledge-based agile methodology. Of the survey participants, only 3% reported that this aspect is not important and only 15% remained neutral, while the remainder of the participants supported the contention that it should be considered in a methodology. Therefore, based on this feedback from the participants (Table 5 and Figure 6), it is reasonable to include the factor of “knowledge engineering and management” in the ASSF.

Table 5. Knowledge engineering and management in agile methods

Knowledge Engineering	Response	Percentage
Very important	7	21%
Important	14	42%
Minimal important	6	18%
Neutral	5	15%
Not important	1	3%

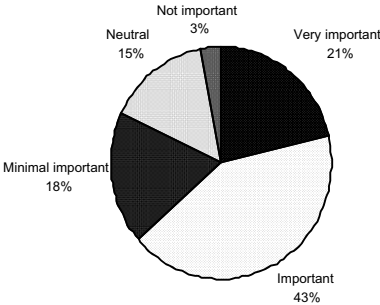


Fig. 6. Knowledge engineering and management in agile methods

3.5 Business Value and Agile Software Development Methodology

In this section, the fifth question has been analysed. The question is: What do you think about the alignment of the business values and agile practices?

ASSF includes and explicitly highlights the aspect of a business value agile-oriented methodology, which has not been discussed in any of the established software development methodology metamodels. Here, in this survey, opinions of the research participants are analysed and presented (Table 6 and Figure 7). Similarly to the previous aspect, only 3% of the survey participants reported that this aspect is not important and only 15% remained neutral, the rest of the participants supporting the argument that it should be considered in a methodology. It seems reasonable, therefore, to include it in the ASSF.

Table 6. Business value and agile software development methodology

Business-Agile Value	Response	Percentage
Very important	15	45%
Important	10	30%
Minimal important	2	6%
Neutral	5	15%
Not important	1	3%

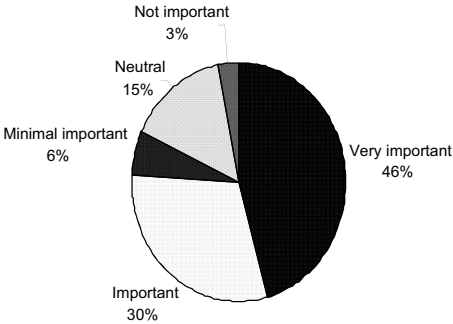


Fig. 7. Business value and agile software development methodology

3.6 Business Policy Impact on Agile Software Development Methodology

In this section, the sixth question has been analysed. The question is: How important is the impact of business policies, goals, strategies and culture in agile software development adoption, execution and governance?

Table 6. Business policy impact on agile software development methodology

Business Policy	Response	Percentage
Very important	22	67%
Important	8	24%
Minimal important	2	6%
Neutral	1	3%
Not important	0	0%

According to the responses of this research question, the consideration of the business impact in ASSF is evident in the responses of the research participants. The participants reported and, interestingly, the data show (Table 7 and Figure 8) that there is not a single participant who opposed this aspect. Therefore, it is argued here that business at large drives the adoption, execution and governance of an agile software development approach.

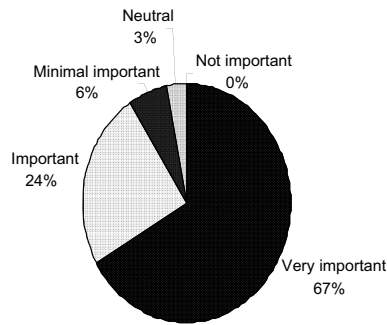


Fig. 8. Business policy impact on agile software development methodology

4 Validity and Limitations

Construct validity is concerned with the assessing whether or not the survey instrument (questionnaire) that is used represents the attributes or aspects being measured. The included aspects of the ASSF have been taken from the initial research. The responses from the participants show that all the attributes and aspects can be considered relevant for representing the domain of software development methodologies. However, their importance in this context has been determined on the basis of the responses received from the research participants.

External validity is concerned with the generalization of the results to other contexts and environments than the one in which the study was conducted [15]. External validity was examined by distributing and conducting the survey with 33 research participants (from medium to large size organizations), from ten different countries, at conferences as well as in research and software development organizations.

There are some limitations, which we think quite reasonable and worth mentioning. A disadvantage of this survey questionnaire, which may be considered here, is that the research participants were provided with a list of possible choices (most of the time) and asked to select from that list. This pre-emption may limit the focus of the research participants to the aspects or attributes of the ASSF that have been reported and used in this questionnaire. However, we tried to address this issue by providing an extra option of “others” wherever possible and appropriate in the question. This allowed the research participants to specify any new aspects or attributes, which they might think should be considered in the context of this research. It was also possible that the research participants may misinterpret the questions provided in the questionnaire; therefore, an administrative questionnaire was used in this survey to explain the meanings of the questions to the participants, if required. Another issue, which is worth considering, is that the responses of the participants are based on and limited to their personal opinion, knowledge, beliefs and attitudes regarding the various aspects of software development methodologies. This situation may cause problems when participants’ perceptions may be inaccurate or when aspects or attributes identified as important for software development method may not in fact be important at all. However, similar to many other opinion-based research studies (for example, [1, 2, 7, 8]), we have full confidence that the findings of this research are based on the

data that have been collected from the research participants, who have been involved and have vastly diversified experience in the research, engineering and implementation of software development methodologies. The sample size of the research participants may be another concern since the data collection, due to the time constraint in this research, restricted further data collection so that only the original 33 participant responses could be collected and analysed. To get a broader view on this research topic, and to make the results of this research more general, more time, participants and organizations need to be involved. However, a true effort has been put in to make the research findings more general and, therefore, participants from ten different countries, both from research (18 participants) and industry (15 participants) sectors were included in this research.

5 Conclusion

This paper has presented the analyses of the opinions of 33 research participants in order to determine the relevance and importance of the elements of the ASSF, which are used to represent the concepts or aspects of an agile software development methodology. The main elements, which have been presented in this paper, may be classified into: process, people, agility, abstraction, product, tools, agile governance, agile knowledge and business (business value and policy). The relevance and importance of these elements have been analysed.

The aspects of people, process, tools and product have already been established and highlighted in the various metamodels of software development methodologies (for example ISO/IEC 24744 [6]). However, along with these already well recognized aspects, this analysis revealed the relevance and importance of the five newly identified aspects of a software development methodology in this research that have not been explicitly considered previously: the aspects of choice, such as agility, abstraction, governance, knowledge and business. The response of the participants on the importance of these all aspects justifies the inclusion of these aspects in the ASSF.

ASSF, containing these elements, may be used to construct agile or non agile or hybrid knowledge-based, governance-based, business-value driven software development methodologies for various abstraction mechanisms, such as for object-oriented, agent-oriented and service-oriented etc. The analysis findings that have been presented in this paper will be discussed further in our future research.

References

1. Baddoo, N., Hall, T.: Motivators of software process improvement: An analysis of practitioner's views. *Journal of Systems and Software* 62, 85–96 (2002)
2. Beecham, S., Tracy, H., Austen, R.: Software Process Problems in Twelve Software Companies: An Empirical Analysis. *Empirical Software Engineering* 8, 7–42 (2003)
3. Brinkkemper, S.: Method engineering: engineering of information systems development methods and tools. *Information and Software Technology* 38(4), 275–280 (1996)
4. Coolican, H.: *Research Methods and Statistics in Psychology*. Hodder and Stoughton, London (1999)

5. Henderson-Sellers, B., Qumer, A.: Using method engineering to make a traditional environment agile. How agile is agile? *Cutter IT Journal* 20(5), 30–37 (2007)
6. ISO/IEC: Software Engineering: Metamodel for Development Methodologies. ISO/IEC 24744. International Standards Organization/International Electrotechnical Commission, Geneva, Switzerland (2007)
7. Niazi, M., Babar, M.A.: De-motivators of Software Process Improvement: An Analysis of Vietnamese Practitioners' Views. In: Münch, J., Abrahamsson, P. (eds.) PROFES 2007. LNCS, vol. 4589, pp. 118–131. Springer, Heidelberg (2007)
8. Niazi, M., Wilson, D., Zowghi, D.: Critical Success Factors for Software Process Improvement: An Empirical Study. *Software Process Improvement and Practices Journal* 11(2), 193–211 (2006)
9. Paetsch, F., Eberlein, A., Maurer, F.: Requirements Engineering and Agile Software Development. In: Proceedings of the IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, Linz, Austria, pp. 308–313. IEEE Computer Society, Los Alamitos (2003)
10. Qumer, A., Henderson-Sellers, B.: Measuring agility and adoptability of agile methods: A 4-Dimensional Analytical Tool. In: Guimarães, N., Isaias, P., Goikoetxea, A. (eds.) Procs. IADIS International Conference Applied Computing, Spain, pp. 503–507. IADIS Press (2006)
11. Qumer, A.: Defining an integrated agile governance for agile software development environments. In: Concas, G., Damiani, E., Scotto, M., Succi, G. (eds.) XP 2007. LNCS, vol. 4536, pp. 157–160. Springer, Heidelberg (2007)
12. Qumer, A., Henderson-Sellers, B.: An evaluation of the degree of agility in six agile methods and its applicability for method engineering. *Inf. Software Technol.* 50(4), 280–295 (2008)
13. Qumer, A., Henderson-Sellers, B.: A framework to support the evaluation, adoption and improvement of agile methods in practice. *Journal of Systems and Software* 81, 1899–1999 (2008)
14. Qumer, A., Henderson-Sellers, B.: An agile toolkit to support agent-oriented and service-oriented computing mechanisms. In: Münch, J., Abrahamsson, P. (eds.) PROFES 2007. LNCS, vol. 4589, pp. 222–236. Springer, Heidelberg (2007)
15. Regnell, B., Runeson, P., Thelin, T.: Are the Perspectives Really Different-Further Experimentation on Scenario-Based Reading of Requirements. *Empirical Software Engineering* 5(4), 331–356 (2000)
16. Taylor, J.L.: Lightweight Processes for Changing Environments. *Dr. Dobbs's Journal* 27(11), 82 (2002)
17. Williams, L., Cockburn, A.: Agile Software Development: It's about Feedback and Change. *Computer* 36(6), 39–43 (2003)

Facilitating Inter-organisational Collaboration via Flexible Sharing of Rapidly Developed Web Applications

Ioakim (Makis) Marmaridis*, Xufeng (Danny) Liang, and Athula Ginige

School of Computing and Mathematics, University of Western Sydney
Sydney, Australia

makis@scm.uws.edu.au,

danny@scm.uws.edu.au,

a.ginige@uws.edu.au

<http://www.uws.edu.au>

Abstract. Increased competitiveness in business requires organisations to work with each other to extend their reach. To enable flexible and nearly ad-hoc information sharing and collaboration we have advocated the use of Dynamic eCollaboration. There are three types of information that must be shared: publicly accessible, information provided via online services, and finally, parts of or the entire web application. To facilitate the rapid pace of Dynamic eCollaboration, web applications must be created quickly and the sharing mechanism should be driven by end-users. In this paper we are presenting the integration of the Bitlet framework for web-based information sharing with our work on the Smart Business Object (SBO).

1 Introduction

Information sharing on the web is one of the cornerstones of eCollaboration [12] and the need for it is increasing as the growth in this area of research indicates [3,4,5,6]. To that end, many approaches and tools already exist for sharing information on the web. Unfortunately, existing approaches and tools are limited and unable to cater for anything beyond the sharing of publicly accessible information. In practice, there is a strong need for people to be able to share information that is not always publicly available, collaborate using web applications jointly, and easily aggregate the shared information. These are key features of information sharing that eCollaboration depends on.

As eCollaboration is gaining popularity as a business practice, it is necessary for information sharing approaches to keep up with the user requirements. At the same time, it is not uncommon to require applications that are purposely built to support collaborative work. These application must also support not only the specifics of the business logic, but also collaborative access and work distribution. In this paper, we showcase the combination of our Bitlets with

* Corresponding author.

the Smart Business Object (SBO). The Bitlet framework [7] facilitates sharing of web-accessible information including parts, or entire, web applications. The Smart Business Object [8] framework caters for rapid creation of web applications by end-users. The combination shows how applications can be created very quickly and get jointly used in the context of collaboration between different partners whose staff are geographically separated. Also in this paper, we present a walk-through example of an application that we have built and used recently in sharing support material between a number of researchers at our and a collaborating institution.

2 Need for Sharing Web Applications

Throughout our research in the area of flexible organisational eCollaboration that we also refer to as Dynamic eCollaboration, it is shown that collaboration is a key ingredient to organisational success [9], [10]. The ability to perform Dynamic eCollaboration is limited in part by how well businesses can share information with one another. The degree of sharing is in turn constrained by current approaches to information sharing. We therefore argue that a new approach to information sharing can introduce additional capabilities for organisations subsequently facilitating Dynamic eCollaboration better. To that end, we established through our research experience a set of key features that information sharing in the context of Dynamic eCollaboration must possess. We then reviewed the different approaches to information sharing against those features looking at the gaps that exist. Finally, building upon that knowledge, we created a new approach for information sharing on the web, based on Bitlets, which incorporates the key features identified previously. Now through this publication we extend the toolset available to organisations practising Dynamic eCollaboration via allowing not only sharing of existing web based applications but also the rapid creation of new ones as well if required.

2.1 Features of Information Sharing in Dynamic eCollaboration

In our experience to date, effective information sharing within the context of Dynamic eCollaboration must offer users control, flexibility and choice. In the combination of Bitlets with SBO we strive to maintain these abilities and continue empowering the end-users. The table below summarises the list of features we see as necessary for effective information sharing in Dynamic eCollaboration.

2.2 General Principles of Sharing Information on the World Wide Web

The most widely exploited feature of the World Wide Web when it comes to information sharing is that each element found online (also known as a resource) can be uniquely identified and accessed via a URL (universal resource location). These URLs were originally devised in order to allow for resources to link to each other (as in page to page links) and also to allow for other types of resources such

Table 1. Features of Information Sharing in Dynamic eCollaboration

Control	Users must be able to initiate sharing of information with one or more other users and maintain the ability to revoke access to that information or amend the terms of sharing at any time.
Flexibility	The information to be shared may come be publicly accessible, accessible as an online service via authentication or directly from an existing or a new, user-created web based application. Users must be able to capture and share information from any of these sources.
Choice	Users must be able to share information that is static or that changes over time. Changing information should be statically, or dynamically.

as images and rich media to be included into pages. Fortunately, this properly of the web can also be exploited when it comes to information sharing. At the most primitive level information sharing can be achieved by simply sharing the URL of a particular resource with others.

Although all sharing methods presented in this paper, including Bitlets, fundamentally rely on URLs for addressing web resources, some of the sharing approaches also allow for additional capabilities such as metadata descriptions, annotations, sharing with context, information aggregation and access arbitration to shared information just to name a few. In our previously published work we have performed a comparative analysis between existing approaches to information sharing and Bitlets [7].

2.3 Limitations of Current Approaches to Sharing

All the current tools and approaches that we reviewed during the course of our research follow the same general architecture pattern shown in Figure 1.

Under this pattern, a user discovers some interesting information on the web and decides to share it. The URL of the page containing that information will then need to be communicated via one of many available methods. The second user is then able to browse to that URL themselves. Upon their first visit to that original web page, they can then bookmark that URL for future reference thus completely bypassing the mechanism that was used to communicate the URL to them in the first place. Also, once the first user has shared the URL to the page he has found to be of interest, he has no ability to revoke this. Depending on the tool he used to communicate the URL he may be able to revoke the notification message but other users may still have the original site bookmarked.

Architecture aside, most of the current tools do not allow for information to be shared maintaining the context of the page it came from intact. From all the solutions reviewed, only copying and pasting from the browser into a document allows for context to be partially maintained (it does not always work as expected and success is dependent on particular browser versions and word processing applications used). A handful of online sharing services exist as well to facilitate inter-personal sharing however even the more elaborate ones also allow for the information page to be captured and stored on their systems however they

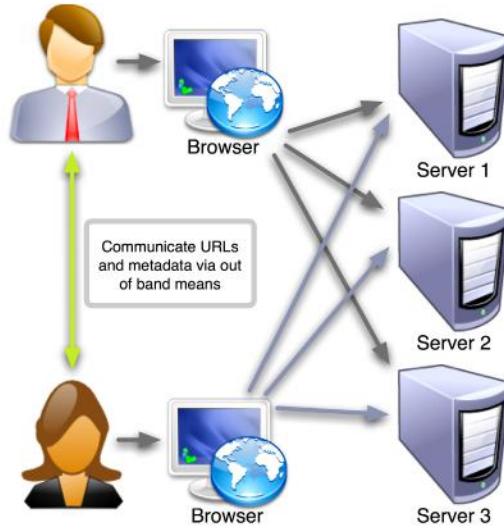


Fig. 1. Generalised Architecture Pattern of Current Information Sharing Approaches

only capture the html content of the page. They leave links to included resources such as images, CSS stylesheets and javascripts unaffected pointing back to the originating site. Therefore, images may still change or even completely disappear overtime. CSS stylesheets may also change to reflect a new site design for instance endangering the readability of the cached html copy. Bitlets come to address all of the limitations above as well as other smaller ones such as allowing different versions of the shared information to be kept if required.

3 Bitlet Framework Overview

3.1 Bitlet Fundamental Architecture Shift

We define a Bitlet as being a stand-alone, sharable bundle of information and associated metadata (such as user provided description and keywords or tags) that can be shared with others over the web. Bitlets encapsulate the entire content of a web page including any referenced resources such as images, media files, style sheets or JavaScript files. Bitlets are designed to be end-user driven and require very few, simple steps to create and work with. Unlike passing a single URL or a URL with a description to another person, a Bitlet encapsulates the page content and other metadata. A Bitlet also offers increased flexibility when it comes to sharing it with others compared to traditional means of sharing. With current methods for information sharing, one typically has to use out of band methods of sharing URLs, documents or other types of pointers to web accessible data. On the other hand, a Bitlet can be selectively shared with individuals or groups of individuals and its creator can at any time revoke access to it. This ability alone offers a great amount of flexibility for end-users and will

be another great motivation for them to use Bitlets where they would normally rely on more traditional sharing methods.

Sharing using Bitlets is fundamentally different from the way all other current approaches operate. This is because Bitlets use a different sharing paradigm. Instead of sharing a pointer (URL) to information residing on a web page somewhere, the entire information and descriptions, keywords and other metadata is shared instead. All access to this Bitlet goes through a single logical path (via the Bitlet Server) and all refreshing of the Bitlet content or interactive use of that content again goes through the same logical path and it is arbitrated. Figure 2 shows a view of the Bitlet architecture.

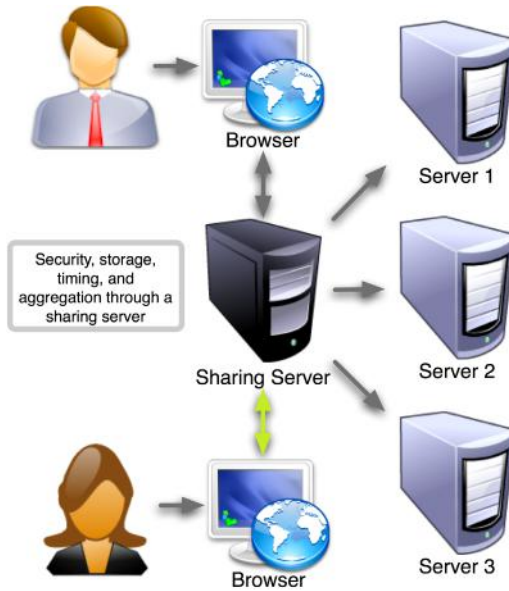


Fig. 2. Bitlet Architecture that Allows Control of Sharing, Aggregation and Two-way Sharing

The key idea here is to arbitrate access to the actual web accessible data by means of the Sharing or Bitlet server. Instead of making the shared data readily available by directly pointing users to it, a Bitlet can instead offer access to the same data, enrich the user experience by offering metadata about the shared data and allow the Bitlet creator control over the sharing throughout the collaboration process. The Sharing or Bitlet server need not be a single physical computer either. It can be distributed across a number of servers for better redundancy and fault-tolerance. Where multiple Bitlet servers are used, they have the ability to communicate with one another and transparently synchronise selected Bitlets for later disconnected (or offline) use.

3.2 Key Advantages and Features

Because of the very different architecture that Bitlet based sharing adopts in comparison to current approaches it is possible to offer a whole host of unique features that other methods cannot match. These features include: Sharing of pages that require authentication - via the Bitlet server impersonating access. Ability to share content interactively (a web form for instance). Maintain full access control of the shared information using role based access control (RBAC) [11], [12]. Share information that is automatically refreshed when required and cache a copy locally. Group multiple Bitlets together into a bundle and present or manage them as a whole.

3.3 The Role of Bitlets in Facilitating Dynamic eCollaboration

Bitlets have been created as a novel method for sharing information in the context of Dynamic eCollaboration, however no amount of technology will be successful in assisting in the adoption of Dynamic eCollaboration unless end-users actively embrace it and make use of it. We strongly believe that Bitlets are an effective method for sharing information on the web and that end-users will be highly motivated to use this over other existing methods. We base this belief in the solid analysis of requirements that we have performed in previous work about what Dynamic eCollaboration needs to be effectively supported and made accessible to organisations of all sizes and its end-users. While we have not yet had the chance to carry out a longitudinal study of how people adopt to the use of Bitlets, there are several key advantages Bitlets have to offer making them quite attractive. These are as follows:

- Trust levels can be better expressed through varying the degree and amount of sharing quickly and easily. All sharing is done from inside the browser. There is no need other system such as email, phone or instant messaging for carrying the sharing metadata out of band.
- Enablement of users to share useful information with others only for the duration of a collaborative project and maintain complete control of the sharing. Unlike sending an email with a URL that once sent is nearly impossible to revoke and even if revoked the recipient still has access to the information since they are accessing it directly via the provided URL.
- Internal systems remain better secured through sharing parts of them via Bitlets. Before Bitlets were available it was not uncommon for people to have to create temporary guest accounts on internal systems of theirs in order to allow collaborating partners to access data from them. With the Bitlet server arbitrating all access to web applications such requirement is waived. Selective pages of a web application can be shared without the creation of new users or security roles in the web application.
- Convenience and speed of setup and use is another key motivating factor we expect to attract users to the use of Bitlets as the preferred sharing method for collaboration. Having to identify upfront what information a business

partner requires and then do the necessary changes to existing web applications and data as to allow such access is very hard to do. Particularly so when collaboration is of short to medium term and information sharing requirements change frequently during the project's lifetime. Bitlets allow end-users to share information as they see fit. No expert IT involvement is necessary in re-configuring existing systems or managing security permissions etc.

Empirically, we are seeing a lot of excitement from our Bitlet users and its increased capabilities in facilitating Dynamic eCollaboration.

4 Advantages of Using SBO for Building Web Applications

When two or more organisations decide to engage in Dynamic eCollaboration, it is not uncommon that their requirements for the project at hand are rather unique and they could require a specialised application to be created in order to support those requirements. It is for these cases where extremely rapid development of simple line of business web applications is needed. The Smart Business Object [8] fills this need.

Smart Business Object is a realisation of the End-User Development (EUD) paradigm (see [13]) for making software “easy to develop”. Using the Smart Business Object framework, data driven web applications can be created in two steps:

1. Model the necessary business objects using SBOML (Smart Business Object Modelling Language), a compact, textual modelling language with near-English syntax
2. Generate different web views of the modelled business objects using the SBO's UI Generation Tool

Modelling Smart Business Objects using SBOML means the generation of the underlying database(s) for object persistence, the ORM (Object Relational Mapping) mapping classes, and a rich portfolio of web user interfaces for presenting the modelled business objects onto the web.

Thus, functional web applications can be easily generated within an extremely condensed timeframe. SBOML has a high-level of abstraction and is based on English lexicon. To model an “employee” business object, for instance, we can literally type the definition (in SBOML expressions) into the SBO Builder tool as shown in Figure 3. In runtime, Smart Business Object is capable of making logical inferences for the most suitable web user interfaces to present the business object's attributes. For example, SBO can automatically renders a Google map for an *address* attribute and displays a *photo* attribute as image (as shown in Figure 4).

Once the business objects are modelled, we can request the rendering of commonly used web *UI Components* in business web applications, such as tables, forms, and charts, for the modelled business objects using the *UI Generation*

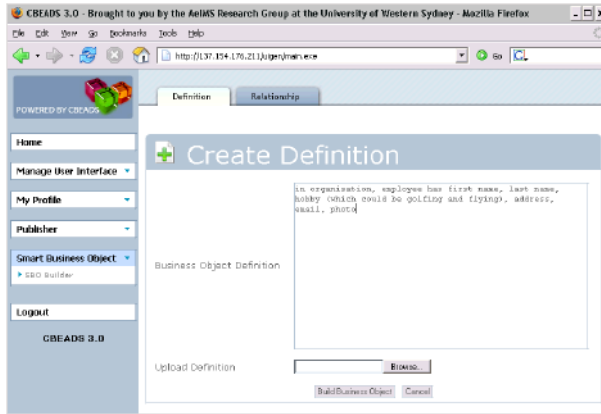


Fig. 3. The SBO Builder

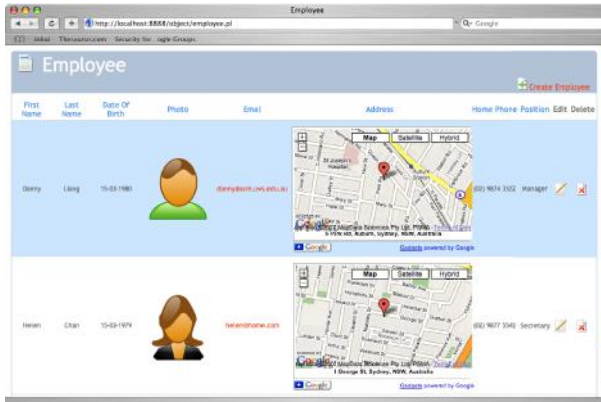


Fig. 4. A Generated Web Application

Tool. The *UI Generation Tool* provides a rich set of options for fine-tuning the generated web user interfaces. Figure 4 is an sample application generated using the toolkit. Consequently, an adept business user with some experience in interacting with web applications can be trained quickly to use the SBO's toolkit to create *instant* web applications.

5 Sharing SBO Based Applications Using Bitlets

These core advantages of combining the Bitlet framework for sharing with SBO is that should new applications need to be created, they do not need to pay any attention to sharing or make any special provisions for supporting collaboration. Instead, they can be built as stand-alone web applications that with the integration of Bitlets can be instantly turned into collaborative applications. Therefore

they can be shared across the network with one or more business partners as it is required. Also the Bitlet framework allows the sharing of web applications and collaboration using those even when these do not exist before hand on each business partner. Only one of the business partners must have the application and they can create bitlets out of it and share those. Bitlets are self-standing and contain all the information required for collaborative use, including the ability to post information back to the originating application or periodically refresh the content of parts of web applications that have been shared.

To showcase the power of integrating Bitlets with SBO-created applications we are describing in this section a recent application that we created for managing research resources of interest between a disparate group of researchers situated in physically different locations. The motivating need was to allow sharing of research material amongst a group of researchers quickly and easily using the web as a medium.

Using this system, researchers that wish to share information with others in their group access CBEADS [14] - a web based framework for developing and deploying web applications - and once they authenticate against it, they can access the Regina environment which is the SBO built application.s server.

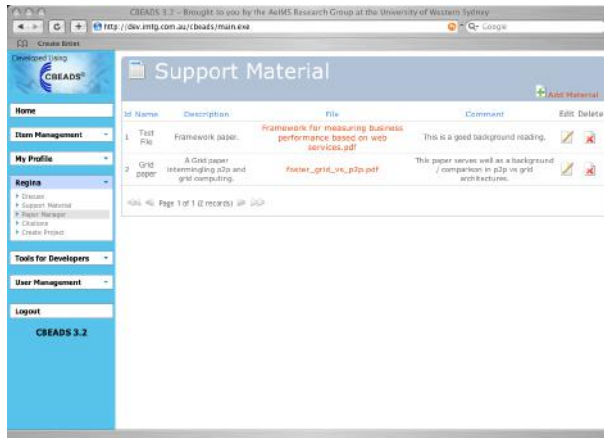


Fig. 5. Main Application Screen - Showing all available research resource materials

This is the main application page and here one can see what resources are available, can also add a new piece of material, edit existing ones or remove selected ones.

To share the uploaded materials, the user can click on the special bookmarklet with the name “Create Bitlet”. This will bring up the details page for creating a new Bitlet and the user provides additional details as shown in Figure 6.

The Mode setting defines whether the Bitlet contents are going to be changing over time or not. Static Bitlets are like a snapshot as at the time they were

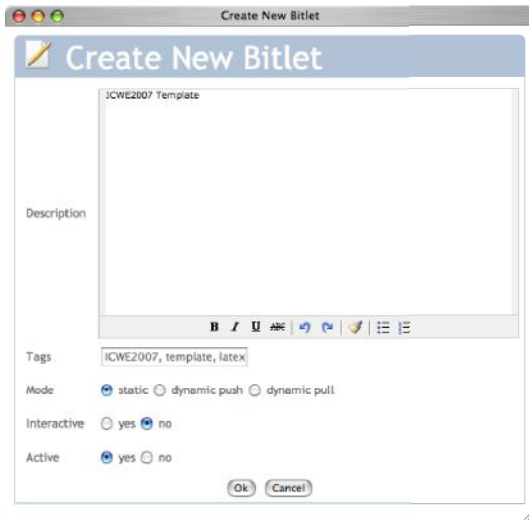


Fig. 6. Create a Bitlet - Make a bitlet out of the main screen of the application

created. Dynamic push means that the Bitlet get automatically updated at regular intervals. Dynamic pull means that the recipient will trigger a Bitlet update every time they open it up. The Interactive setting determines whether a Bitlet allows the recipient to interact with it or not. Non-interactive Bitlets do not allow form post-backs whereas interactive ones do. Without going into implementation details, if the “add material” form was shared as an interactive Bitlet, recipients of it could then fill the form and submit it so that any new materials would be added in the original application.

Once the Bitlet has been created it will automatically become accessible to others in the group who will be able to access the original application through it without having direct access to the application themselves.

Bitlets are agnostic of their content, they can be used to capture publicly available web pages or other web accessible content as long as it is accessible by URL. SBO-created applications are well suited for interacting with Bitlets. SBO applications allow for each and every page that is generated to be addressable via a unique URL, therefore the Bitlet framework can easily distinguish pages and retrieve them in case they are dynamically shared.

Where authentication is required the Bitlet framework is able to perform user impersonation and handle this requirement transparently from the end-user and recipient of the Bitlet. Also, when Bitlets content needs refreshing the Bitlet framework handles this transparently as well.

6 Conclusion

In this paper we have outlined the need for end-user driven, flexible and nearly ad-hoc collaboration via sharing of web applications. We also presented the

increased power that comes from combining Bitlets with Smart Business Object, two complimentary technology frameworks for sharing and rapidly building web applications that fulfil collaboration objectives.

Via the integration of our existing technologies of the Bitlet framework for web information sharing and the Smart Business Object, we have been able to provide an environment for rapid web application creation and sharing that is end-user driven and extensible.

These two key technologies provide a great foundation for collaboration however a lot more is needed for providing end-to-end Dynamic eCollaboration. This paper has not covered some of these additional topics such as virtual teams and issues of trust between collaborating partners. These are covered in some of our work previously published [12][15].

Although what we have presented in this paper offers a complete solution for sharing web based applications between individuals, many interesting areas of research and future work remain available and should be worked on. On the sharing front, Bitlets should eventually be extended to incorporate better aggregation of Bitlets allowing the sharing of information that is derived from a set of Bitlets. Also content tagging and highlighting should be supported within each Bitlet rather than treating them as small black boxes once shared. On the other hand, Smart Business Object should better facilitate the aggregation of business objects in heterogeneous forms, such as those that exist in service-oriented architecture, while providing consistent web user interfaces for transparent access to the aggregated business objects.

References

1. Marmaridis, I., Ginige, J., Ginige, A.: Web based architecture for dynamic collaborative work. In: International Conference on Software Engineering and Knowledge Engineering (2004)
2. Marmaridis, I., Ginige, A.: Framework for collaborative web applications. LNCS, pp. 539–544. Springer, Heidelberg (2005)
3. Ahn, G.J., Mohan, B.: Secure information sharing using role-based delegation. In: Proceedings of the seventh ACM symposium on Access control models and technologies. tY - CONF., pp. 810–815 (2004);
4. Charles, J., Phillips, E., Ting, T., Demurjian, S.A.: Information sharing and security in dynamic coalitions. In: Proceedings of the seventh ACM symposium on Access control models and technologies (SACMAT 2002). SACMAT, pp. 87–96 (2002)
5. Seidmann, A., Sundararajan, A.: Building and sustaining interorganizational information sharing relationships: the competitive impact of interfacing supply chain operations with marketing strategy. In: Proceedings of the eighteenth international conference on Information systems (ICIS 1997), pp. 205–222. Association for Information Systems, Atlanta (1997)
6. Edwards, W.K.: A framework for information sharing in collaborative applications. In: Proceedings of ACM CHI 1994 Conference on Human Factors in Computing Systems, ser. INTERACTIVE POSTERS. Human Factors in Computing Systems, vol. 2, pp. 89–90 (1994)

7. Marmaridis, I., Ginige, A.: Sharing information on the web using bitlets. In: Proceedings of the 6th international conference on Web engineering, pp. 185–192. ACM Press, New York (2006)
8. Liang, X., Ginige, A.: Smart business object - a new approach to model business objects for web applications. In: International Conference on Software and Data Technologies (ICSOFT 2006), vol. 2, pp. 30–39 (2006)
9. Lee, M.: Collaborating to Win - Creating an Effective Virtual Organsiation, Taipei, Taiwan, March 26-27 (2004)
10. Ginige, A., Murugesan, S., Kazanis, P.: A road map for successfully transforming smes into e-businesses. Cutter IT Journal 14 (2001)
11. Tripathi, A., Ahmed, T., Kulkarni, D., Kumar, R., Kashiramka, K.: Context-based secure resource access in pervasive computing environments. In: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops. tY - CONF., pp. 159–163 (2004)
12. Park, J.S., Sandhu, R., Ahn, G.-J.: Role-based access control on the web. ACM Trans. Inf. Syst. Secur. 4(1), 37–71 (2001)
13. Lieberman, H., Paterno, F., Klann, M., Wulf, V.: End User Development, ser. Human-Computer Interaction Series, vol. 9, ch.1. Springer, Heidelberg (2006)
14. Ginige, A.: New Paradigm for Developing Evolutionary Software to Support E-Business. In: Handbook of Software Engineering and Knowledge Engineering, vol. 2, pp. 711–725. World Scientific, Singapore (2002)
15. Marmaridis, I., Ginige, J.A., Ginige, A., Arunatilaka, S.: Architecture for evolving and maintainable web information systems. In: Proceedings of the international resource management association conference, IRMA 2004 (2004)

Sales Forecasting Using an Evolutionary Algorithm Based Radial Basis Function Neural Network

R.J. Kuo¹, Tung-Lai Hu², and Zhen-Yao Chen^{3,*}

¹ Department of Industrial Management
National Taiwan University of Science and Technology
43, Sec. 4, Keelung Rd., Taipei, 10607, Taiwan
rjkuo@mail.ntust.edu.tw

² Department of Business Management
National Taipei University of Technology
1, Sec. 3, Chung-hsiao E. Rd., Taipei, 10608, Taiwan
jameshu@ntut.edu.tw

³ Institute of Industrial and Business Management
National Taipei University of Technology
1, Sec. 3, Chung-hsiao E. Rd., Taipei, 10608, Taiwan
Tel.: +8862-2771-2171 ext. 4540
keyzyc@gmail.com

Abstract. This study intends to present a hybrid evolutionary algorithm for sales forecasting problem. The proposed algorithm is a hybrid of particle swarm optimization (PSO) algorithm and genetic algorithm (GA) for gathering both their merits to improve the learning performance of radial basis function neural network (RBFnn). Model evaluation results of papaya milk sales data show that the proposed algorithm outperforms the sole approach algorithms and traditional Box–Jenkins model in accuracy.

Keywords: Sales forecasting, hybrid evolutionary algorithm, radial basis function neural network, particle swarm optimization, genetic algorithm.

1 Introduction

Box and Jenkins developed the auto-regressive integrated moving average (ARIMA) methodology for forecasting time series events in 1976. Time series data are often examined in hopes of discovering a historical pattern that can be exploited in the forecast [2]. Basically, statistical models have sound theoretical basis and have been successfully applied for industries. However, the limitations of these methods are that they need a previous modeling to be established and a large part of them are developed for specific nonlinear patterns. In other words, they are not capable of modeling other types of nonlinearity in time series. As a result, they are not commonly used for general forecasting purposes [13]. A linear correlation structure is assumed among the time series data and therefore, no nonlinear patterns can be captured by the ARIMA models [33].

* Corresponding author.

To improve forecasting nonlinear time series events, researchers have developed alternative modeling approaches [12]. We have learnt through the literature that radial basis function neural network (RBFnn) can be applied to perform excellent training and approximation for nonlinear function (problem) [27]. In addition, the approaches related to particle swarm optimization (PSO) and genetic algorithm (GA) have been studied for its manner of combination extensively with different hybrid algorithm [19] and proved to have better performance. The previous researchers have adopted the RBFnn structure along with other sole approaches such as PSO and GA to implement the learning of the network. Thus, this study proposes a hybrid of PSO and GA (HPG) evolutionary algorithm for training RBFnn. The proposed algorithm gathers virtues of PSO and GA approaches to ascend learning performance of the network.

This paper supplements a practical application on the historical sales forecasting data of papaya milk to expound the superiority of the proposed HPG algorithm. The results reveal that the sole approaches above mentioned may be combined ingeniously and redeveloped into a hybrid algorithm which aims for appropriate training and adjustment for the parameters of the network, and further obtaining a relatively more accurate forecasting performance than those of the sole approaches.

2 Literature Review

The use of artificial neural network (ANN) for forecasting and modeling has generated considerable interest in recent years [4]. Zou et al compares ANN, ARIMA and the combined models in forecasting the wheat price of China Zhengzhou Grain Wholesale Market, and the results show the ANN model forecasts are considerably more accurate than the traditional ARIMA models, which used as a benchmark [34]. In contrast to ANN, the RBFnn has a more compact topology for learning [17] and it is more robust than the ordinary networks [30]. RBFnn is suited for applications such as pattern discrimination and classification, interpolation, prediction, forecasting, and process modeling [26]. There have been many reports where practitioners have applied RBFs to various time series problems [16].

A typical hidden node in an RBFnn is characterized by its centre, which is a vector with dimension equal to the number of inputs to the node. Gaussian basis functions (Φ) are the most frequently used radial basis functions (RBFs) with the following form [10]:

$$\Phi(\|x - c_j\|) = \exp\left(-\left(\frac{\|x - c_j\|}{\sigma_j}\right)^2\right), \quad (1)$$

where $\|x - c_j\|$ is the Euclidean distance between an input vector x and a centre c_j , and σ_j represents the width of the j^{th} RBFnn hidden node. The key problems of RBFnn, including determining centers and widths of RBF, the number of hidden

nodes, weights between hidden and output layers, and the parameters of hidden layer are optimized locally not globally [3]. Evolutionary computation (EC) techniques are a search algorithm using the concepts of evolutionary pressure to search for fit solutions to problems [32]. The emergence of various neural network (NN) topologies and efficient learning algorithms have led to a wide range of successful applications in forecasting [7].

Owing to its particular structure, a NN is very good in learning using some evolutionary algorithms such as the GA [28] and PSO [10]. The hybrid of GA and PSO (HGAPSO), which incorporates PSO into GA, is proposed and applied to recurrent-network design [18]. Dong focuses on the advantage of PSO into the mutation process of GA for improving the GA learning efficiency [8]. Grimaccia et al. proposed a new hybrid genetical swarm optimization (GSO) approach, consists in a stronger cooperation of GA and PSO, maintaining the integration of them for the entire optimization run [15]. This kind of updating results in an evolutionary process where individuals not only improve their score for natural selection of the fitness or for good-knowledge sharing, but for both of them at the same time [15]. Lee proposed a hybrid GA-PSO approach, that GA and PSO both work with the same initial population [22]. When solving an N dimensional problem, the hybrid approach takes $4N$ individuals that are randomly generated. The $4N$ individuals are sorted by fitness, and the top $2N$ individuals are fed into the real-coded GA to create $2N$ new individuals by crossover and mutation operations. The new $2N$ individuals created from real-coded GA are used to adjust the remaining $2N$ particles by the PSO method. Denoted as GA-PSO, this hybrid technique incorporates concepts from GA and PSO and creates individuals in a new generation not only by crossover and mutation operations as found in GA but also by mechanisms of PSO [22].

3 The Proposed HPG Algorithm

Since PSO and GA both work with a population of solutions, combining the searching abilities of both methods seems to be a good approach [25]. The PSO-based algorithm can simultaneously select the proper number of RBFs and adjustable parameters of the RBFnn with a special fitness function [10]. The GA-based algorithm [28] starts with an initial population of chromosomes, which represent possible network structures and contain the associated center locations. The chromosome that has produced the minimum value of the objective function is selected as the optimum NN model [28].

The proposed HPG algorithm, which combines the evolutionary learning approaches of the PSO and GA, is designed to resolve the problem of network parameter training and solving in RBFnn. New generations are produced using the Roulette wheel selection [14] mechanism and four genetic operators: uniform crossover, exchange (swap) mutation, deletion and addition. The algorithm that stops after a specific number of generations have been completed. We use the root mean squared error (RMSE) to measure the error committed by the algorithm when being used to predict

the instances in the learning set. We have used the inverse of RMSE as fitness function, i.e., $\text{Fitness} = \text{RMSE}^{-1}$ [5]. The fitness value for above mentioned algorithms applied to sales forecasting are computed by maximizing the inverse of RMSE defined as [22]:

$$\text{Fitness} = \text{RMSE}^{-1} = \sqrt{\frac{N}{\sum_{j=1}^N (y_j - \hat{y}_j)^2}}, \quad (2)$$

where y_j is the actual output and $\hat{y}_j$ is the predicted output of the learned RBFnn model for the j^{th} training pattern. Next, the nonlinear function that the RBFnn hidden layer adopts is the Gaussian function shown in formula (1). Then, the optimal values of parameters solution can be obtained and used in the algorithm with the network to solve the problem for sales forecasting.

3.1 The Dissect of the HPG Algorithm

The algorithm simultaneously implements different learning approaches aimed at the first and the last half of the initialized primitive population. The first half of population is implemented with Maximum selection PSO learning [10] (PSO approach), the last half of population is implemented with GA learning (GA approach). When the algorithm proceeds to the next step, it will implements GA and Maximum selection PSO learning approaches for the first and the last half of the population.

For population in PSO approach, in which particle figures out the best solution after consulting itself and other particles, and decides the proceeding direction and velocity. Thus, executing an evolutionary computation through the PSO approach would obtain an enhanced evolution population, which is better than the initial population. Meanwhile, the population found by PSO approach learning would preserve better vector solution for the next generation due to the memorial property of PSO information sharing.

Due to the property of global search with GA approach, no matter what the fitness values of the chromosomes in population are, they all have the chances to proceed with uniform crossover operator and enter the next generation of population to evolve. In this way, it meets the spirit of GA approach and ensures the genetic diversity in the future evolution process, and proceeds to obtain a new enhanced population. Through some steps such as crossover, mutation, and addition/deletion, the GA approach may further help individuals to share the parameter values solution with other individuals in population. Thus, the solution space in population could be changed gradually and converge toward the optimal solution.

To continue, the algorithm will form [PSO+GA] and [GA+PSO] sub populations respectively, and combine as a new generated population. In the above mentioned procedures, the whole population will gather the strength of PSO and GA approaches during evolutionary learning to make better training for two sub populations to continue preceding afterward evolution. The pseudo code for the proposed HPG algorithm is illustrated as Fig. 1.

```

Begin
    Initialize population;
    Calculate the weights and fitness values of chromosomes;
    Preserve the best chromosome;
    for (i = 0 ; i < number of generations ; i++) {
        GA ( FIRST_HALF ); // run GA approach with the first half of population
        PSO ( FIRST_HALF ); // run PSO approach with the first half of population
        PSO ( LAST_HALF ); // run PSO approach with the last half of population
        GA ( LAST_HALF ); // run GA approach with the last half of population
        Perform uniform crossover operator on chromosomes;
        Perform two-point mutation operator on chromosomes;
        Perform addition or deletion operators on chromosomes;
        Calculate the weights and fitness values of chromosomes;
        Select a new population from both [PSO+GA] and [GA+PSO] populations
            with Roulette wheel selection;
        Replace the weakest chromosome with the previous best chromosome;
        Preserve the best chromosome;
        Decrease linearly operations of the updated rates of  $P_c$ ,  $P_m$ , and  $k$  by degrees
            with the number of generation;
    }
End
PSO Learning:
Begin
    Calculate the local and global optimal values;
    Perform Maximum selection type PSO learning method;
    Update velocities and positions of particles;
    Calculate the weights and fitness values of particles;
End

```

Fig. 1. The pseudo code of the proposed HPG algorithm

4 Experiment Results and Comparison

This research precedes practical short-term sales forecasting analysis of papaya milk. The daily sales data of 500 cm³ containers of papaya milk were offered by a parent company of chain convenience stores in Taiwan. The analysis has assumed that the influence of external experimental factors does not exist, and the sales trend for papaya milk is not interfered by any special events. The results are compared to others sole approach algorithm and traditional ARIMA models, illustrating the accuracy of the proposed HPG algorithm.

All these numerical results were performed on a PC with Intel XeonTM CPU, running at 3.40GHz symmetric multi-processing (SMP), and 2GB of RAM. Simulation were programmed in the Java 2 Platform, Standard Edition (J2SE) 1.5. Additionally,

EViewsTM software was also used the analysis of ARIMA models to calculate the numerical results. This study elaborates how data is input to RBFnn for prediction through above mentioned algorithms, and comparison with ARIMA models. Furthermore, a comparison will be carried among these prediction results.

4.1 Experimental Parameters Setting

There are several the values of parameters within RBFnn that must be set in advance to perform training for application of forecasting analysis. It is better than trial and error way in literature that determines the appropriate the values of parameters from the verified domain to train RBFnn to implement sales forecasting through above mentioned algorithms. Above mentioned algorithms start with parameters setting shown in Table 1 and are meant to ensure consistent basis in this application.

Table 1. Parameters setting for above mentioned algorithms

Description	Value
Population size	30
The maximum number of the RBFnn hidden node center	0
The number of the RBFnn hidden node centers	[1, 50]
The widths of RBFnn hidden layer	[1000, 40000]
Inertia weight	0.5
Acceleration constants	2
Crossover rate (Uniform crossover)	.5
Mutation rate (Exchange mutation)	0.05
Addition and Deletion rate	0.005
The maximum number of generations (epochs)	1,000

90% observations will be used as learning set, the remaining 10% will be treated as forecasting set [34] and were used for validation and one-step-ahead forecasting. We applied the observations from January-1 1995 to January-14 1996 as the in-of sample set (including 95 testing data) with 341 observations for learning purpose and the remainder as the out-of-sample set with 38 observations for forecasting purpose. For confidentiality reasons, the observations are linearly normalized between zero and one. Formula (3) [34] is used in this study:

$$X'_i = \frac{(X_i - X_{\min})}{(X_{\max} - X_{\min})}, \quad (3)$$

where X'_i , X_i , $X_{\max}$, and $X_{\min}$ are the normalized value of the observed sales data, actual value of the observed sales data, maximum observation value of the sales data and the minimum observation value of the sales data respectively.

4.2 Building ARIMA Models

This research carries out sales forecasting based on ARIMA models, the procedures can be divided into the following stages [25]:

1. **Data Identification:** This research precedes the data identification of ARIMA models through augmented Dickey-Fuller (ADF) testing [6]. The results reveal that the sales data are stationary and thus differencing is not necessary. Thus, we can adopt ARMA (p, q) (i.e., ARIMA ($p, 0, q$)) models to precede estimation and forecasting of sales data.
2. **Model Estimation:** We eliminated the data that the coefficient of each parameter item are insignificantly different from zero, and then sequentially sift the candidate ARMA models out. Akaike information criterion (AIC) [1] criteria were employed to sift the optimal model out [9]. From the results, we can infer that the AIC value (i.e., -2.30622) of ARMA (1, 2) is the smallest among all candidate ARMA models, revealing that it is the optimal model and thus the most appropriate one for the daily based sales forecasting.
3. **Model Diagnosis:** This study adopts Q-statistic (i.e., Ljung-Box statistic) [21] to measure the residual values of ARMA (1, 2) model whether white noise (i.e., serial non-correlation). The results of model diagnosis reveal that the values of Q-statistic are greater than 0.05 in result of ARMA (1, 2) model, in which the result is serial non-correlation and it had been suitable fitted.
4. **Model Forecasting:** The research adopted the fittest ARMA (1, 2) model, which had verified model estimation and diagnosis to proceed forecasting on historical sales data of papaya milk.

4.3 The Error Measures of the Sales Forecasting Performance

The root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) are the most commonly used error measures in business, and are used to evaluate the forecast models [4] in this paper. Additionally, for the time series methodology the MAPE is shown calculated on the prediction dataset [7]. In the research, three criteria (i.e., RMSE, MAE, and MAPE) were used to compare the sale forecasting performance of the HPG algorithm against other methodologies, i.e., the PSO-based algorithm [10], GA-based algorithm [28], and the ARMA (1, 2) model.

The sales forecasting results of forecasting set are shown in Fig. 2. The forecasting performances of above mentioned methodologies with papaya milk historical sales data are present in Table 2. Among these methodologies, the results derived from RMSE, MAE, and MAPE of the HPG algorithm were the smallest ones.

This study applies a new hybrid algorithm to the problem of forecasting the daily sales of papaya milk. The main objective of this paper is to present a hybrid evolutionary algorithm for sales forecasting with high precision using RBFnn. The proposed HPG algorithm is applied to perform better adjusting and learning on RBFnn parameters solution. Additionally, comparisons with other algorithms based on sole evolutionary approaches, as well as the traditional ARIMA models, are also discussed, with an attempt to achieve the most precise results from the sales forecast through the algorithm.

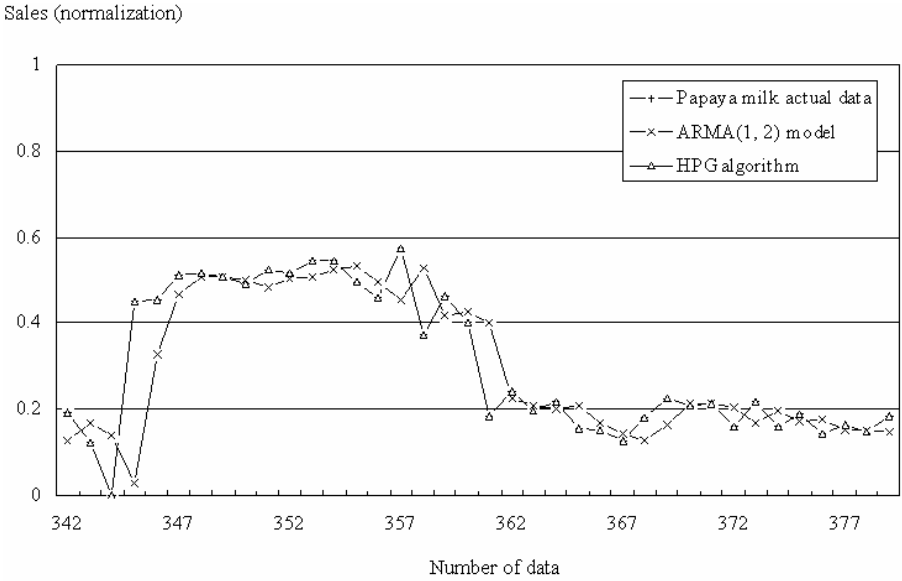


Fig. 2. The forecasting results for different methodologies using papaya milk sales data

Table 2. The forecasting errors for different methodologies using papaya milk sales data

Error Methodology	RMSE	MAE	MAPE (%)
PSO [10]	7.31E-12	5.98E-12	25.7E-10
GA [28]	69.40E-12	53.60E-12	192E-10
HPG	0.474E-12	0.357E-12	1.38E-10
ARMA (1, 2)	0.09455	0.05557	18.9163

5 Conclusions

The study for the proposed HPG algorithm provides the settings of some parameters, such as hidden node centers, widths, and weights of RBFnn. An example of application on sales data forecasting of papaya milk using the algorithm with RBFnn trained has been given. It has been shown that the proposed hybrid algorithm performs more efficiently than the sole PSO and GA approach based algorithm. The results proved that compared to others sole approach algorithm and traditional ARIMA models, the algorithm has better parameter setting of network and consequently to enables RBFnn to perform better learning and sales forecasting.

References

1. Akaike, H.: A new look at the statistical model identification. *IEEE Trans. on Automat. Control* AC. 19, 716–723 (1974)
2. Box, G.E.P., Jenkins, G.: *Time Series Analysis, Forecasting and Control*. Holden-Day, San Francisco (1976)

3. Chen, S., Mei, T., Luo, M., Yang, X.: Identification of Nonlinear System Based on a New Hybrid Gradient-Based PSO Algorithm. In: International Conference on Information Acquisition, Jeju City, Korea, pp. 265–268 (2007)
4. Co, H.C., Boosarawongse, R.: Forecasting Thailand's rice export: Statistical techniques vs. artificial neural networks. *Computers & Industrial Engineering* 53, 610–627 (2007)
5. delaOssa, L., Gamez, J.A., Puetra, J.M.: Learning weighted linguistic fuzzy rules with estimation of distribution algorithms. In: IEEE Congress on Evolutionary Computation, pp. 900–907. Sheraton Vancouver Wall Centre Hotel, Vancouver (2006)
6. Dickey, D.A., Fuller, W.A.: Likelihood Ratio Statistics for Autoregressive Time Series with A Unit Root. *Econometrica* 49(4), 1057–1072 (1981)
7. Doganis, P., Alexandridis, A., Patrinos, P., Sarimveis, H.: Time series sales forecasting for short shelf-life food products based on artificial neural networks and evolutionary computing. *Journal of Food Engineering* 75(2), 196–204 (2006)
8. Dong, H.K.: GA-PSO based vector control of indirect three phase induction motor. *Appl. Soft. Comput.* 7(2), 601–611 (2007)
9. Engle, R.F., Yoo, B.: Forecasting and Testing in Cointegrated Systems. *Journal of Econometrics* 35, 588–589 (1987)
10. Feng, H.M.: Self-generation RBFNs using evolutionary PSO learning. *Neurocomputing* 70, 241–251 (2006)
11. Francq, C., Makarova, S., Zakoian, J.M.: A class of stochastic unit-root bilinear processes: Mixing properties and unit-root test. *Journal of Economics* 142, 312–326 (2008)
12. Ghiassi, M., Saidane, H., Zimbra, D.K.: A dynamic artificial neural network model for forecasting time series events. *International Journal of Forecasting* 21, 341–362 (2005)
13. Ghiassi, M., Saidane, H.: A dynamic architecture for artificial neural networks. *Neurocomputing* 70, 397–413 (2005)
14. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization & Machine Learning*. Addison-Wesley, Reading (1989)
15. Grimaccia, F., Mussetta, M., Zich, R.E.: Genetical Swarm Optimization: Self-Adaptive Hybrid Evolutionary Algorithm for Electromagnetics. *IEEE Transaction on Antennas and Propagation, Part 1* 55(3), 781–785 (2007)
16. Harpham, C., Dawson, C.W.: The effect of different basis functions on a radial basis function network for time series prediction: A comparative study. *Neurocomputing* 69, 2161–2170 (2006)
17. Huang, C.M., Wang, F.L.: An RBF Network With OLS and EPSO Algorithms for Real-Time Power Dispatch. *IEEE Transactions On Power Systems* 22(1), 96–104 (2007)
18. Juang, C.F.: A hybrid of genetic algorithm and particle swarm optimization for recurrent network design. *IEEE Trans. Syst. Man Cybern., Part B: Cybernetics* 34(2), 997–1006 (2004)
19. Kao, Y.T., Zahara, E.: A hybrid genetic algorithm and particle swarm optimization for multimodal functions. *Applied Soft Computing* 8, 849–857 (2008)
20. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: IEEE International Conference on Neural Networks, pp. 1942–1948. IEEE Service Center, Perth (1995)
21. Kmenta, J.: *Elements of Econometrics*, 2nd edn. Macmillan Publishing Co, New York (1986)
22. Lee, Z.J.: A novel hybrid algorithm for function approximation. *Expert Systems with Applications* 34, 384–390 (2008)
23. Lu, C.F., Juang, C.F.: Evolutionary fuzzy control of flexible AC transmission system. *IEE Proc. Gener. Transm. Distrib.* 152(4), 441–448 (2005)

24. McLeod, A.I., Zhang, Y.: Faster ARMA maximum likelihood estimation. *Computational Statistics & Data Analysis* 52, 2166–2176 (2008)
25. Ong, C.S., Huang, J.J., Tzeng, G.H.: Model identification of ARIMA family using genetic algorithm. *Applied Mathematics and Computation* 164, 885–912 (2005)
26. Ordieres, J.B., Vergara, E.P., Capuz, R.S., Salazar, R.E.: Neural network prediction model for fine particulate matter (PM_{2.5}) on the US-Mexico border in El Paso (Texas) and Ciudad Juarez (Chihuahua). *Environmental Modelling and Software* 20, 547–559 (2005)
27. Salajegheh, E., Gholizadeh, S.: Optimum design of structures by an improved genetic algorithm using neural networks. *Advances in Engineering Software* 36, 757–767 (2005)
28. Sarimveis, H., Alexandridis, A., Mazarakis, S., Bafas, G.: A new algorithm for developing dynamic radial basis function neural network models based on genetic algorithms. *Computers and Chemical Engineering* 28, 209–217 (2004)
29. Shi, Y.: Eberhart, R.C.: Parameter Selection in Particle Swarm Optimization. In: Porto, V.W., Waagen, D. (eds.) EP 1998. LNCS, vol. 1447, pp. 591–600. Springer, Heidelberg (1998)
30. Tsai, C.H., Chuang, H.T.: Deadzone compensation based on constrained RBF neural network. *Journal of the Franklin Institute* 374, 361–374 (2004)
31. Valenzuela, O., Rojas, I., Rojas, F., Pomares, H., Herrera, L.J., Guillen, A., Marquez, L., Pasadas, M.: Hybridization of intelligent techniques and ARIMA models for time series prediction. *Fuzzy Sets and Systems* 159, 821–845 (2008)
32. Whigham, P.A., Recknagel, F.: An inductive approach to ecological time series modeling by evolutionary computation. *Ecological Modelling* 146, 275–287 (2001)
33. Zhang, G.P.: Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing* 50, 159–175 (2003)
34. Zou, H.F., Xia, G.P., Yang, F.T., Wang, H.Y.: An investigation and comparison of artificial neural network and time series models for Chinese food grain price forecasting. *Neurocomputing* 70, 2913–2923 (2007)

Micro Implementation of Join Operation at Clustering Nodes of Heterogenous Sensor Networks

Ehsan Vossough¹ and Janusz R. Getta²

¹ School of Computing and Mathematics
University of Western Sydney,
Campbelltown, Australia

² School of Computer Science and Software Engineering,
University of Wollongong,
Wollongong, Australia

Abstract. A *wireless sensor network* is an information technology infrastructure capable of processing theoretically unlimited sequences of data commonly called as *data streams*. A *heterogeneous sensor network* consists of *sensing nodes* creating the streams of atomic data items and *clustering nodes* joining several data streams into one stream of complex data items and performing the preliminary data processing. This work investigates the microprogramming implementation of join operation at the clustering nodes. We define a data stream join operation and we prove that it can be computed in an incremental way. Then, we propose a number of optimisation techniques for its processing. The micro implementation of join operation on 8-bit microprocessor controller with memory and interfacing ports is derived from the incremental processing algorithm. We describe architecture of firmware implementation of join operation and we propose a network implementation of a number of join modules. The experimental results include testing of join operation on two and many data streams.

Keywords: Sensor network, join operation, data stream, clustering data streams, incremental processing, optimization.

1 Introduction

The advanced technologies of electronic sensing and wireless communication enable transmission and processing of large amounts of data collected from the wireless sensor networks monitoring the wide geographical areas over the long periods of time. A wireless sensor network [1,2] is an information technology infrastructure that can be used for the continuous processing of theoretically unlimited sequences of data. Such sequence of data is commonly called as a *data stream*. The streams of data are created through the continuous observations performed in the physical world, biological systems, financial systems, or information technology systems. A data stream is an unbound, continuously expanding, and storage unlimited sequence of elementary or complex data items obtained either directly from the sensor nodes or from the clustering and processing nodes of a wireless sensor network [3].

The continuous monitoring of complex systems with sensor networks may provide important information about the properties and behaviour of such systems. The typical applications include online analysis of environmental data, monitoring and surveillance of complex engineering systems, and wide ranges of medical applications. For instance, sensor networks enable the continuous observations and processing of weather parameters, monitoring the behavior of wild animals, monitoring drinking water quality, disaster relief management, structural monitoring of buildings and bridges, measuring human body parameters [2], and many others.

1.1 Data Stream Join Operation

A heterogeneous sensor network consists of two types of nodes: *sensing nodes* and *clustering nodes*. The functionality of *sensing node* is limited to collecting numerical data from the sensors and performing the relatively simple computational tasks on these data due to the restricted speed of an embedded processor and limited transient memory.

These limitations can be reduced through "clustering" of the data streams obtained from the sensing nodes into a *single* stream of complex data items and later on through processing of a complex stream at a more powerful node. Clustering of data streams can be used to eliminate redundant data from the closely located sensors, to filter out the incorrect values and to group data collected within a short period of time. As each sensor node contributes to a single data stream, clustering reduces the total number of streams processed by a sensor network. Clustering of data streams requires the implementation of an operation commonly called as a *data stream join operation* [4]. The operation takes its name and some of its semantics from the *relational algebra join operation*. In the relational database model, a join operation combines the rows from two or more relational tables. The relational algebra join operation can be defined as a Cartesian product of two relational tables followed by *filtering* over a formula of propositional calculus.

A *data stream join operation* [5] acts on the fast and continuously changing finite subsets of data streams later on referred to as *windows*. In this work we consider a data stream join operation defined as a join of n windows preceded and followed by filtering, and sorting of data items in an output window. The elementary actions performed by the operation filter out its input streams, combine the data items into n -tuples, filter the n -tuples, and finally sort the result into an output stream.

A view of a sensor network as a distributed data stream processing system justifies the existence and implementation of *data stream join operation* [16]. It is needed in sensor networks where the sensing nodes are not powerful enough to perform a complete analysis of data and they have to rely on data from the adjacent nodes in the network. Then, data stream join operation can be used to cluster the streams of elementary data items into a stream of complex items and to send it to more computationally powerful nodes.

1.2 Clustering Nodes in Heterogenous Sensor Networks

A typical *homogeneous sensor network* consists of a number of identical *sensing nodes*. Each node has the capabilities to measure a given physical parameter and to

perform local data processing with an embedded 8 bit, 16 MHz processor and 4k of transient memory. A sensing node is powered by a local and possibly solar powered battery that transmits and/or receives data [2] serially.

The application of *clustering nodes* [2] increases the processing power of *heterogeneous (multilevel) sensor networks*. In this sort of networks sensing nodes collect data, verify data correctness, keep and maintain a window on n most up to date elementary data items, and transmit the updates to a clustering node. Clustering nodes join a number of data streams obtained from sensing nodes into a stream of complex data items and transmit the output stream for processing at a mobile personal class computer. A clustering node has a more powerful processor, more transient memory, and more powerful battery.

1.3 Implementation of Join Operation at a Clustering Node

A clustering node in a heterogeneous sensor network is "the right location" for the implementation of *data stream join* operation. As sensing nodes continuously transmit the elementary data items to a clustering node it has the most up-to-date information about the values of parameters measured by the sensing nodes. Due to theoretically unlimited length of data stream and finite amount transient memory available at a clustering node, the computations should be performed on the finite subsets of selected data items from each stream, i.e. on *the windows on data streams*. Originally, the windows are kept at the sensing nodes. A sensing node stores and maintains one window in its transient memory, which contains n most recent data items. Each time a new item is collected from a sensor, it is appended to a window and the oldest measurement is removed from the window.

As each sensing node keeps in its transient memory the most up to date state of a window, the join of n data streams from n nodes is equivalent to a relational algebra join of n windows. Each time one of the windows changes its contents, a relational algebra join operation must be recomputed on all windows involved in data stream join. Such approach is very ineffective, as it repeats almost the same processing after every single modification of a window. It is known that some of the operations on data streams can be computed in *an incremental way*. It means that the latest result of an operation is stored and it is used together with the most recent modification of a window to compute the next result. We show in Section 3, that it is possible to perform the computations of data stream join operation in *an incremental way*.

A clustering node stores a new result of join in a window located at the node. Next, the contents the window passes through filtering and sorting. Finally, the data items included in a result of join operation are time-stamped and sent to the final processing nodes.

This work investigates an efficient implementation of data stream join operation at a clustering node of heterogeneous sensor network. In order to consider all data items created by every sensing node in a network we have to recompute a join operation on every state of the windows. We expect that implementation of clustering nodes at firmware level will be able to reach the performance appropriate for handling many data streams at a single node. We also find how the computations of join operation can be improved through the incremental data processing. We assume that pre-processed data may be stored on a single module or shared among several nodes until

sent to a nearby clustering node. Single thread architecture of each node eliminates the need for synchronization until a time when data is received. At this point all data arriving from several nodes are synchronized on a clustering node. Generally, volume of processed data on a network is less than the raw data and is less susceptible to errors from network interference with reduced load shedding.

1.4 Structure of the Paper

The paper is organised in the following way. The next section reviews the latest work on the implementation of data stream join operation in homogeneous sensor networks. Section 3 includes the definition of the basic concepts, it shows that join operation can be computed in an incremental way, and it also describes possible ways how the computations of join operation can be optimised. In Section 4, we describe a micro implementation of join operation. It is followed by the presentations of firmware implementation of join operation and network implementation of several join modules in the Section 5. We present the experimental results in Section 6. Finally, Section 7 concludes the paper.

2 Previous Works

The formal systems of operations on data streams have been proposed in [5] and [6]. Both systems included a *relational join* operation adapted to a data stream environment with no implementation proposed. Early implementations were based on a simple model of data streams with no sensor networks involved.

The latter model considered a binary join operation acting on two streams or more precisely, on two finite subsets of the streams, also called as *windows on data streams*. Later, the same approach was extended to join n data streams at once. The *Fjords* architecture [7] aimed at managing multiple queries over a traffic sensor network and used a software implementation of data stream join operation at the sensor proxy nodes. *W-join* operation [8] followed as a more advanced implementations of a data stream join operation. Another technique of joining data streams [9] implemented the operation as a *relational multi-way join* to handle many data streams at a time. An approach proposed in [10] implemented data stream join operation on the data streams that have different arrival rates.

A data stream based approach to the implementation of join operation in a sensor network has a few serious problems. Firstly, as number of sensor nodes increases, the cardinality of multi-way join operation grows to an extent that cannot be handled efficiently by a single node in a network. Secondly, it incurs significant communication costs when all sensor nodes must transmit their data to a single node in the network. When some of the sensing nodes are not able to reach the "join" node, data is re-transmitted through the other sensing nodes. This increases transmission time, consumes more power, and creates additional synchronization problems. Finally, data stream based implementation of join operation is not appropriate for homogeneous sensor networks where the processing power of each node is usually too low to handle complex multi-way join operation.

Another approach for the implementation of a join operation in sensor networks is to consider a network as a system of *distributed* databases. Then, each node in a network is treated as a small database systems and computation of data stream join operation is almost identical to processing a relational join operation in distributed relational databases. This idea [11] is applied with the distributed hash and index-based algorithms that implement range-join operation in sensor networks. A new approach proposed in [12] speeds up the processing by firstly joining the synopses of the data streams in order to eliminate the tuples that do not contribute to join results. An important problem in processing data stream join operation is to find which sensor nodes in the homogeneous network should process the data and which sensor should transmit the data. This problem is addressed in [13] where a cost-based model is used to select the best join strategy for a given query. The experiments conducted in this work confirm that, dynamic selection of the join strategy based on the cost model is always better than fixed strategy. In [14] a solution to the same problem shows that different in-network approaches are theoretically applicable and it determines a suitable time when a superior strategy is to perform the join at a specialized processing node. Many approaches proposed so far have assumed that the amount of available memory at nodes is known in advance and it is large enough to buffer a subset of join arguments. A research work [15] proposed a distributed join algorithm that considers the memory limitations at nodes and does not make a priori assumptions on the available memory at the processing nodes. Recently, [16] proposed an optimal algorithm for implementation of a join operation in dense sensor networks that minimizes the communication costs. The algorithm adopts a well know nested-loop relational implementation of join operation to a sensor network environment. A general approach to the implementation of data stream processing applications in sensor networks based on a formal model presented in [5] been recently presented in [17].

To our best knowledge there were no attempts to implement a join operation in heterogeneous sensor networks and directly at a microprogramming level of the clustering nodes.

3 Formal Backgrounds

This section includes the definition of basic concepts such data stream, window on a data stream, homogeneous, and heterogeneous sensor networks, data stream processing application and join operation. Next, we show how to compute in an incremental way a data stream join operation on n windows and we present the optimisation techniques that can be applied while processing the operation.

3.1 Basic Concepts

A *data stream* is a continuously growing sequence of data items. It is defined as an infinite sequence of data items $d_1, d_2, d_3, \dots$.

Let S be a set of data streams. An *application* processing the streams in S is defined as n -argument function $f: S \times S \times \dots \times S \rightarrow S$ that takes n streams as the arguments and returns a data stream as its value. For example, an application $filter_\phi(s)$ removes from a stream s all data items that do not satisfy a condition ϕ and returns a stream of items ordered in the same way as in s . An application $max(s)$ returns a stream that

consists of one element, i.e. the largest element in a stream s . An application $join_{r,a=s,b}(r,s)$ when applied to the relational streams r and s returns a stream of pairs of values (x,y) such that x comes from a stream r and y comes from a stream s and such that $x.a=y.b$. The data items included in the result of the operation are output in no particular order. An application $join-ts_{r,t \sim 0..1} s.t(r,s,r.t)$ creates the pairs (x,y) , $x \in r$ and $y \in s$ such that value of timestamp $x.t$ is in range of $0..1$ of a value of timestamp $y.t$ and order of items in an output stream is determined by the values of timestamp $x.t$.

If we consider a data stream s as an infinite sequence of data items then processing of an application $f(s)$ would take an infinite amount of time to complete. To avoid this problem, we observe the intermediate results of the processing $f(s^{(1)})$, $f(s^{(2)})$, ..., $f(s^{(n)})$, $f(s^{(n+1)})$, ... over an infinite period of time, where $s^{(1)}$, $s^{(2)}$, ..., $s^{(n)}$, $s^{(n+1)}$, ... are the finite subsequence of stream s valid at the moments $t^{(1)}$, $t^{(2)}$, ..., $t^{(n)}$, $t^{(n+1)}$, ... and such that $s^{(1)} \subseteq s^{(2)} \subseteq \dots \subseteq s^{(n)} \subseteq s^{(n+1)} \dots$. It means, that we replace the infinitely long computations of an application $f(s)$ with an infinite sequence of partial results $f(s^{(1)})$, $f(s^{(2)})$, ..., $f(s^{(n)})$, $f(s^{(n+1)})$ where the next state of a stream is obtained from the concatenation of the most recently arrived data items $\delta^{(i+1)}$ with a previous state of a stream, i.e. $s^{(i+1)} = s^{(i)} \cdot \delta^{(i+1)}$.

The computations of $f(s^{(i+1)})$ can be simplified in some cases. If a result of $f(s^{(i+1)})$ is equal to the result of $f(s^{(i)} \cdot \delta^{(i+1)})$ and it is possible to find a function $g(x,y)$ such that $f(s^{(i)} \cdot \delta^{(i+1)}) = g(f(s^{(i)}), f(\delta^{(i+1)}))$ then it is possible to compute $f(s^{(i+1)})$ incrementally as an application of g to the previous results $f(s^{(i)})$ and recently arrived data items $\delta^{(i+1)}$, i.e. $f(s^{(i+1)}) = g(f(s^{(i)}), f(\delta^{(i+1)}))$.

In order to further speed up the processing of $f(s^{(i+1)})$ we usually take a practical approach where there is no need to process all data items from the current state of a stream. Instead, we assume that older data items from the current state of a stream do not have an important impact on the results of an application and there is no need to process these items in every cycle of the continuously running application. Then, an inclusion of $s^{(i)}$ into $s^{(i+1)}$ no longer holds and instead of $s^{(i)}$ we consider a window $w^{(i)}$. When a state $s^{(i)}$ of a stream changes into $s^{(i+1)}$ then a new window $w^{(i+1)}$ is created from the most recently processed window $w^{(i)}$. The new window has the oldest data items $\delta^{(i)}$ removed from one side and the newest data items $\delta^{(i+1)}$ added to the other side. This sort of windows satisfies a condition $\delta^{(i)} \cdot w^{(i+1)} = w^{(i)} \cdot \delta^{(i+1)}$, see Figure 1, where "dot" operation denotes a concatenation of two sequences of data items.

A value of $f(w^{(i+1)})$ is equal to $f(\delta^{(i)} \rightarrow w^{(i)} \cdot \delta^{(i+1)})$ where $\delta^{(i)} \rightarrow w^{(i)}$ denotes elimination of a sequence $\delta^{(i)}$ from the beginning of window $w^{(i)}$. If, additionally, it is possible to find a function $h(x,y,z)$ such that $f(\delta^{(i)} \rightarrow w^{(i)} \cdot \delta^{(i+1)}) = h(f(\delta^{(i)}), f(w^{(i)}), f(\delta^{(i+1)}))$ then $f(w^{(i+1)})$ can be computed in an incremental way.

$$f(w^{(i+1)}) = g(f(\delta^{(i)}), f(w^{(i)}), f(\delta^{(i+1)})) \quad (1)$$

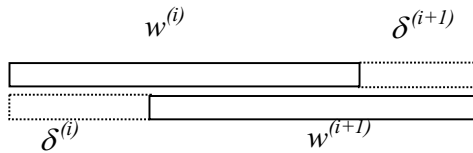


Fig. 1. Modifications $\delta^{(i)}$ and $\delta^{(i+1)}$ of a window $w^{(i)}$

A data stream application $f(s_1, \dots, s_n)$ outputs a stream of data items as an outcome of the continuous invocations of f over the continuously changing n windows $w_1, \dots, w_n$ over the streams $s_1, \dots, s_n$. A convenient pattern of processing of n streams is to first individually filter out each stream, to assemble the results into one stream, and to finally process the complex data items. Such model of processing is convenient in the environments where the devices generating the streams of data have the limited computational abilities due to limited power, e.g. in the sensor networks where data items can be processed locally and later on sent to a central site for aggregation and more complex processing. For example, an environmental system may generate the streams of measurements of temperature, air pressure, and humidity. The pre-processed data are later on assembled into the triples like $\langle \text{temperature, air-pressure, humidity} \rangle$ that include the different measurements from more or less the same moment in time. Assembling the streams $s_1, \dots, s_n$ of simple data items into a stream of complex data items is performed by a *data stream join* operation $join(s_1, \dots, s_n, \phi, \rho)$ defined as

$$join(s_1, \dots, s_n, \phi, \rho) = sort_{\rho}(filter_{\phi}(s_1 \times \dots \times s_n)) \quad (2)$$

A selection condition ϕ is applied to the result of $s_1 \times \dots \times s_n$, and ρ is a sequence of identifiers of data items that determine an order in which the output stream is formed. The cross product operations $s_1 \times \dots \times s_n$ consider the streams as the sets of data items and return a set of n -tuples, which is later on filtered out with a *filter* operation. In practice, the operation continuously processes a cross product of finite windows $w_1 \times \dots \times w_n$ on data streams.

3.2 Incremental Processing of Join Operation

In a window based approach to processing of infinite streams a data stream join operation has to be continuously reprocessed on n windows $w_1 \times \dots \times w_n$. Each time a window w_k , $k=1, \dots, n$ changes its state through the elimination of data items δ^i and addition of data items in δ^{i+1} an operation $join(w_1, \dots, w_k, \dots, w_n, \phi, \rho)$ must be recomputed.

We shall show that the operation as it is defined in (2) satisfies an equation (1) and because of that it can be computed incrementally. Consider a modification of window w_k such that $w_k^{(i+1)} = \delta^i \rightarrow w_k^{(i)} \cdot \delta^{i+1}$. Then, $join(w_1^{(i)}, \dots, w_k^{(i+1)}, \dots, w_n^{(i)}, \phi, \rho) = join(w_1^{(i)}, \dots, (\delta^i \rightarrow w_k^{(i)} \cdot \delta^{i+1}), \dots, w_n^{(i)}, \phi, \rho)$. The replacement of the right hand side with a definition of *join* operation given in (2) provides an equation:

$$join(w_1^{(i)}, \dots, w_k^{(i+1)}, \dots, w_n^{(i)}, \phi, \rho) = sort_{\rho}(filter_{\phi}(w_1^{(i)} \times \dots \times (\delta^i \rightarrow w_k^{(i)} \cdot \delta^{i+1}) \times \dots \times w_n^{(i)})) \quad (3)$$

An expression $w_1^{(i)} \times \dots \times (\delta^i \rightarrow w_k^{(i)} \cdot \delta^{i+1}) \times \dots \times w_n^{(i)}$ is equal to

$$((w_1^{(i)} \times \dots \times w_k^{(i)} \times \dots \times w_n^{(i)}) - (w_1^{(i)} \times \dots \times \delta^i \times \dots \times w_n^{(i)})) \cup (w_1^{(i)} \times \dots \times \delta^{i+1} \times \dots \times w_n^{(i)})$$

The operations *filter* and *sort* are distributive over the set operations of difference and union. Hence,

$$\begin{aligned}
& join(w_1^{(i)}, \dots, w_k^{(i+1)}, \dots, w_n^{(i)}, \phi, \rho) = \\
& (sort_{\rho}(filter_{\phi}(w_1^{(i)} \times \dots \times w_k^{(i)} \times \dots \times w_n^{(i)})) - sort_{\rho}(filter_{\phi}(w_1^{(i)} \times \dots \times \delta^{(i)} \times \dots \times w_n^{(i)}))) \\
& \cup (w_1^{(i)} \times \dots \times \delta^{(i+1)} \times \dots \times w_n^{(i)}) \quad (4)
\end{aligned}$$

Finally we obtain:

$$\begin{aligned}
& join(w_1^{(i)}, \dots, w_k^{(i+1)}, \dots, w_n^{(i)}, \phi, \rho) = \\
& ((join(w_1^{(i)}, \dots, w_k^{(i)}, \dots, w_n^{(i)}, \phi, \rho)) - join(w_1^{(i)}, \dots, \delta^{(i)}, \dots, w_n^{(i)})) \\
& \cup join(w_1^{(i)}, \dots, \delta^{(i+1)}, \dots, w_n^{(i)}) \quad (5)
\end{aligned}$$

An equation (5) above means, that after a modification of a window $w_k^{(i)}$, a new result of *join* operation $join(w_1^{(i)}, \dots, w_k^{(i+1)}, \dots, w_n^{(i)}, \phi, \rho)$ can be computed from the most recent result of *join* operation $join(w_1^{(i)}, \dots, w_k^{(i)}, \dots, w_n^{(i)}, \phi, \rho)$ by removing the results of *join* obtained from the remaining windows and data elements $\delta^{(i)}$ removed from the window and by adding new results of *join* operation on the remaining windows and data elements $\delta^{(i+1)}$ added to the window. An equation (5) justifies the continuous computations of *data stream join* operation as the repetitions of itself after each modification of any data stream being an argument of the operation.

3.3 Optimization of Join Operation

A significant improvement in the performance of join is possible due to the commutativity of cross product operation. An expression $w_1^{(i)} \times \dots \times \delta^{(i)} \times \dots \times w_n^{(i)}$ can be transformed into an expression $(\dots ((w_1^{(i)} \times \delta^{(i)}) \times w_2^{(i)}) \dots \times w_n^{(i)})$. It means that results of $w_1^{(i)} \times \delta^{(i)}$ can be used in a product with $w_2^{(i)}$ and so on. In a consequence there is no need to record the intermediate results when computing join operations over the modifications and the computations can be performed in a "pipeline" style where the modifications computed by one operation become the arguments of the next operation.

Another kind of optimisation of join operations is possible due to the distributivity of *filter* operation over a cross product operation. An expression $sort_{\rho}(filter_{\phi}(\dots (w_1^{(i)} \times \delta^{(i)}) \times w_2^{(i)}) \dots \times w_n^{(i)}))$ can be transformed into $sort_{\rho}((\dots (filter_{\phi}(w_1^{(i)}) \times filter_{\phi}(\delta^{(i)})) \times filter_{\phi}(w_2^{(i)}) \dots \times filter_{\phi}(w_n^{(i)})))$. A *filter* operation applied before a cross product operation reduces the size of the arguments.

As a simple example, consider a hypothetical implementation of *join* operation that integrates a sensory stream of sample voltages $sv_1(s_1\#, v_1\#, date-time)$ with information about a sensor $s_1\#$ and voltage value $v_1\#$. Similarly, a sensory stream $sv_2(s_2\#, v_2\#, date-time)$. We expect, that *join* operation will create a stream of voltage group pairs $sv_{12}(s_1\#, s_2\#, v_1\#, v_2\#, date-time)$ is the sensory values over the last 1000 samples with date-time taken as the latest of both time stamps [7]. A window w_i over *sample* streams contains 1000 of the most recent values. Each time a new value is finalised and it is inserted into the window, the oldest sample is removed from the window.

A window w_p over sv_1 contains information about the changing voltage values from sensor₁. A window is updated in such a way that whenever a new voltage value from sensor₁ is sampled a *voltage* attribute for respective data item is updated and the previous *voltage* and $v_1\#$ is removed from the window. A data stream join operation $join(sample, voltage, sv_1.v_1\# = sv_2.v_2\#, sample.date-time)$ returns 1000 values of the attributes $v_1\#, v_2\#, date-time$. Only modifications of a window w_i trigger a data stream join operation. The modifications of a window w_p are only recorded in the window and old values are saved in a stream *samples* outside a window. A data stream join operation given above can be implemented as the following expression: $sort_{date-time}(filter_{sv_1.v_1\# = sv_2.v_2\#}(w_i \times w_p))$. In order to implement the incremental computations described by an equation (5) we have to consider a modification $\delta = (\delta^-, \delta^+)$ of a window w_i where δ^- is a set of items removed from the window and δ^+ is a set of items added to the window. The efficiency of the computations depend on the fast evaluation of an expression $filter_{sv_1.v_1\# = sv_2.v_2\#}(\delta \times w_p)$. During the evaluation the contents of δ is scanned sequentially and the contents of w_p is accessed through an index on $sv_1.v_1\#$. An index is used to find a data item in w_p that has the same value of an attribute $v_1\#$ as in a data item selected from delta. If a data element taken from delta belongs to δ^- then joined item is marked with (-) to denote that it should be deleted from the latest result. Otherwise, a new joined item is marked with (+) to denote that it has to be added to the last result. Index on w_p over $v_1\#$ speeds up access to the data items in w_p . Only one pass through the data items in δ is needed and the index is traversed, as many times as many data items are included in δ .

4 Micro Implementation of Join Operation

Microprocessor architecture for a *join* operation described in this section is derived from equation (5), which describes the process for a sliding window. Incremental processing is performed in the transient memory. Upon arrival of a new tuple, it is first inserted into a designated window for that stream and at the same time the oldest tuple is removed from the window. The new tuple is then joined with tuples in windows designated for other streams and the results are inserted into a results list that is called a *bucket*, and at the same time the previous results from the oldest tuple are removed from this bucket.

A brief description of the hardware system intended for our experiments follows. The hardware design utilizes an 8-bit micro processor controller with memory and interfacing ports. This arrangement is referred to as a *module*. Test data streams were generated randomly onboard each module, with values in a specific range of those generated by a sensor. Due to the single threaded operation of modules, a new tuple is completely processed before another new tuple can resume. Results of join are stored onboard the module memory and later sent to the PC. Each data element from a sensor is concatenated with a sensor identification number and a unique timestamp before it leaves the module. At the receiving end packets are grouped together in the right sequence according to their time stamps.

5 Firmware Implementation of Join Operation

PC for our experiments is a desktop dual core AMD 4200 CPU (11,975 MIPS) with 2 Giga Bytes of memory. Figure 2 shows the core of a module called an *MDFLY* development board, which accommodates an 8-bit high performance, low power, Atmel 128 RISC central processor unit (CPU-16MIPS). Program code for the CPU is compiled with an AVR Studio 4 compiler program that is serially delivered and stored onboard the module's non-volatile FLASH memory. The volatile static RAM accommodates sliding windows, temporary storage for program variables, and buckets to store results of stream operations. A unique bucket is allocated for each stream to facilitate searching for tuples and improve performance. The final result is the concatenation of all the buckets that is sent to the PC. Allocation of tuples in a bucket depends on the availability of an empty space (hole) created by removal of an expired tuple, otherwise it is inserted at the top of the bucket.

In the event of an overflow, a new tuple is discarded and a byte containing the bucket ID and tuple ID is sent to the PC. Due to the uniform distribution of the data values in each stream the size of a bucket stays fairly uniform as it is constantly updated by the results.

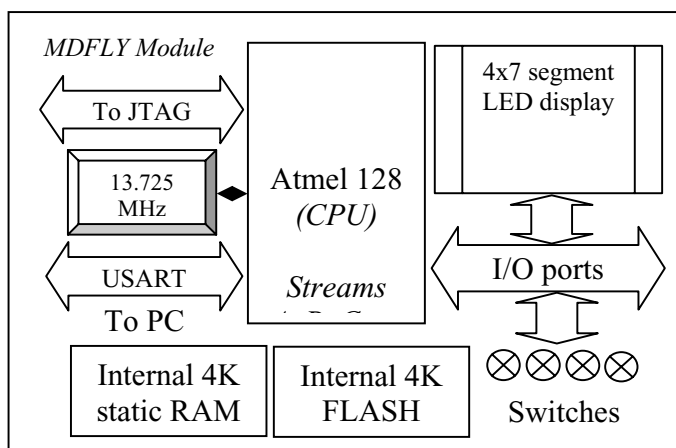


Fig. 2. Basic architecture of an MDFLY development board module

Each module communicates with the PC using two communication interfaces. Firstly, compiled code from the PC is sent to the module's flash via a fast parallel interface. This interface is also used for several other purposes; to reset/start the program on the module, debug the program at various breakpoints, and display critical window and bucket data values. Secondly, a serial interface sends error codes generated at runtime, values at the start and the completion of each experiment and other critical runtime data to the PC. Communication rate for this interface was set to 19200 bits/sec.

A resident program on the PC performs housekeeping and other duties. It updates module's parameters, initializes the module's window and bucket sizes, and records

starting and the completion time for each test. Test duration is measured to within one millisecond.

Since module architecture is single threaded, it is not necessary to synchronize streams in a single module. However, with several remote modules working together, synchronization must be acknowledged when results are materialized in a remote module or a PC. Lowest time stamp algorithm can be applied for this purpose [18, 19]. In the future developments, a Cross Bow encrypted wireless network with inbuilt sensors will cover reasonably large distances depending on availability of power and local battery capacity.

6 Experimental Results

We used a single module to join several streams with some reservations. Inter module communication delays between stream tuples were ignored and tuples arriving from several nodes were joined on the same module. Data values ranged from 1 to 255 and

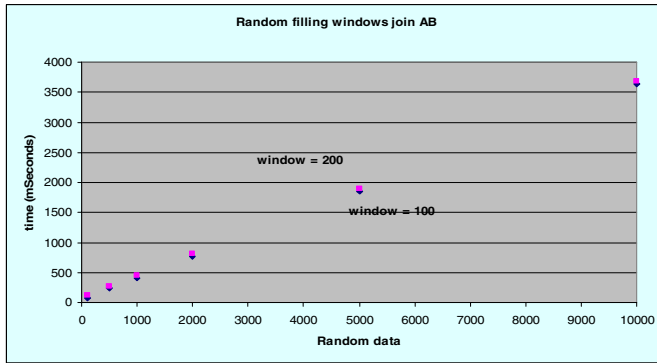


Fig. 3. Populating windows for two streams without joining

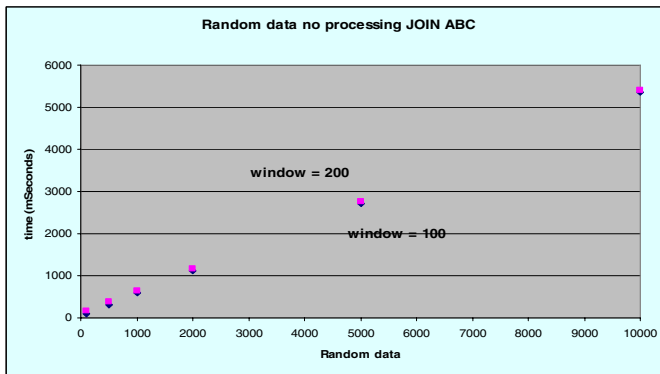


Fig. 4. Populating windows for three streams without joining

sliding windows were tested for two different sizes of 100 and 200 tuples, respectively. Tests included the overhead associated with populating a window without processing the data, joining the streams, removing the expired tuples from a window and the removal of the expired joined tuples from the corresponding buckets, and finally sending the results to the PC. Figures 3 and 4 show the overhead for populating a window. This overhead is a small fraction of the join time, as can be seen in the later tests. Searching and replacement of expired tuples, contributed to a larger proportion of the overhead.

6.1 Joining Two Streams

Figures 5 and 6 show the results for joining two streams. The linear performance is compared with the same code running on a PC which outperformed the RISC CPU by a factor of 420 times. However, a module has an advantage of low power consumption, small size, cheaper construction, long battery life, and solar powered. By employing several modules that process many nodes, there is a considerable reduction in

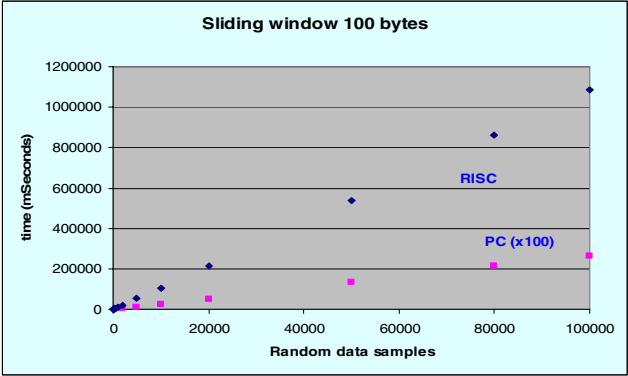


Fig. 5. Joining two streams with window size of 100, compared with a PC

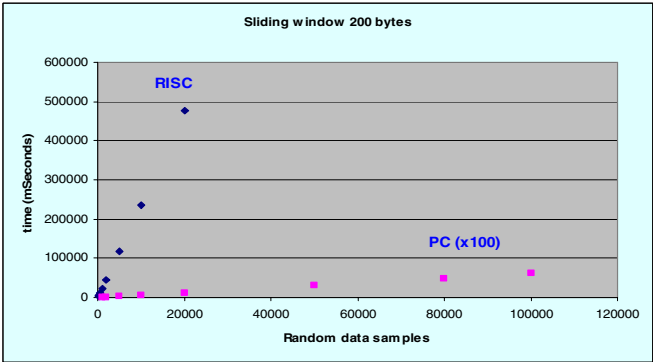


Fig. 6. Joining two streams with window size of 200, compared with a PC

data sent to the PC for the final processing. Modules also share resources among each other and pre process as much data as possible utilizing their CPU idle time. Removal of one module due to loss of power or otherwise does not impede the operation of the remaining modules.

Transmission time for sending contents of a bucket at 19200 bits per second was about 42 milliseconds. This is small overhead that is performed at much larger intervals compared with the time needed to perform several join operations.

6.2 Joining Several Streams

Joining several streams is implemented by joining a new tuple from one stream over windows from other streams in any order. This follows the commutative rule for a join which cannot be applied to a *difference* or another similar operation that is non commutative. A new tuple is fully processed before a next tuple arrives, with a conservative assumption that no tuple is dropped in the process. For example, to join 4

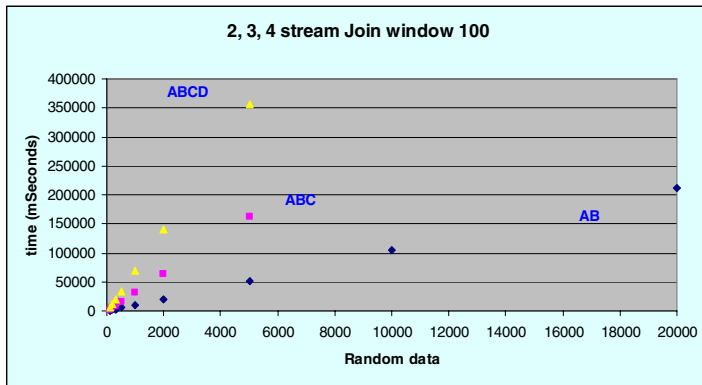


Fig. 7. Joining several streams over a window of 100

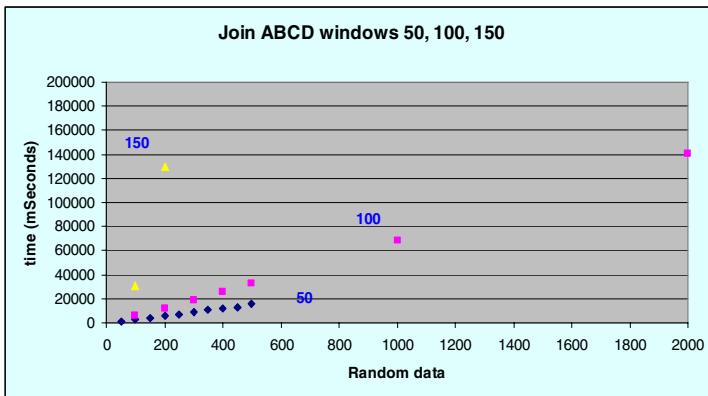


Fig. 8. Joining 4 streams over window sizes 50, 100, 150

streams, a tuple for stream A is joined with windows B, C and D, and results populate the bucket for stream A. Buckets are then sent collectively to PC at regular intervals for processing. If additional processing is performed by the PC on several unprocessed streams, then stream synchronization would have to be implemented [18, 19]. Figure 7 compares the results of joining 2, 3 and 4 streams, respectively, over a window of 100 tuples. Joining four streams is seven times slower than two streams and three times slower than three streams. Performance degradation with windows larger than 50 or more than two streams is relatively nonlinear, as indicated in Figure 8.

In Figure 9, four streams are processed with a single module and 2,000 tuples in the stream. Performance was the same as processing 30,000 tuples with a single PC. Alternatively, 15 modules would be needed to process 4 streams in order to get the same throughput as a PC. The justification in section 6.1 for using modules in place of a PC is equally applicable. One factor that we did not included in the measurements is the sharing of resources among several modules, utilizing the CPU idle time and inter communication between the modules. Module throughput could be enhanced by changing the window sizes where possible at runtime, favouring the smallest window size for different sensors and different conditions. There is a marked performance lag between a window of 100 and 150 tuples but no so much performance degradation between a window of 50 and 100 tuples. If a smaller window size could not be utilized then one resolution would be to allow load-shedding and use approximation methods to reinterpret the final results. A major overhead in these experiments is searching and removing expired tuples in a bucket and in the window. A general solution is to sign all the expired tuples and remove them at a later convenient time. However, due to the limited size of module's onboard memory, a bucket can fill up very quickly necessitating the immediate removal of some expired tuples before more could be inserted. Searching and cleanup time is reduced by approximately 75% by allocating a separate bucket for every stream, rather than searching a single bucket that contains the collective results from all streams.

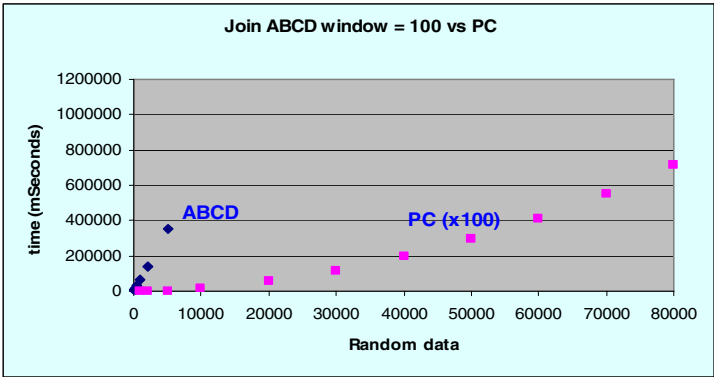


Fig. 9. Comparison of joining four streams with a PC

7 Summary

In this paper we proposed a technique for joining data over a network of electronic sensors that may also include a wireless communication. The system would allow collection and transmission of large amounts of data from the monitoring of wide area networks over long periods of time. Generally, sensors that collect data from weather temperature, ground vibrations, robotic applications, and other blue tooth interfaces would collectively need a large bandwidth, and can be processed locally before sending essential information to a PC for further processing. Close groups of sensors may be hard wired together to a module which may operate on a solar charged battery, would run for months with minimal maintenance. Hardware techniques proposed here aim at reducing the load burden at the data processing level by pre processing streams at front end modules that operate with a single or possibly more CPUs. Networking is implemented by interconnecting several modules and then synchronizing the collected data streams on a central PC. Many types of joining can be performed within a single module or as a collectively shared process among several modules; controlled by a central monitoring station. Latter may be implemented as a single PC or a pre programmed control module. Methods of synchronization suggested are based on time stamp methods that were suggested in previous research. It is anticipated that front end modules will be miniaturized in the future as small battery/solar operated hand held devices that collect, process and send large data streams to remote stations for processing. Implementation of new and innovative joining techniques is our objective for the future research.

References

- [1] Sohraby, K., Minoli, D., Znati, T.: *Wireless Sensor Networks*. John Wiley & Sons, Inc., Chichester (2007)
- [2] Cordeiro, C.M., Agrawal, D.P.: *Ad Hoc & Sensor Networks*. World Scientific Publishing, Singapore (2006)
- [3] Babcock, B., Babu, S., Datar, M., Motwani, R., Widom, J.: Models and issues in data stream systems. In: Popa, L. (ed.) *Proceedings of the Twenty-first ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pp. 1–16. ACM Press, New York (2002)
- [4] Elmasri, R., Navathe, S.B.: *Fundamentals of Database Systems*, 4th edn. Pearson Education, London (2004)
- [5] Getta, J.R., Vossough, E.: Optimization of data stream processing. In: *SIGMOD Record*, vol. 33(3), pp. 34–39 (2004)
- [6] Dani, A., Getta, J.R.: A System Of Operations On Sliding Windows. In: *International Conference on Computing & Communication, ADCOM 2006*, December, pp. 595–600 (2006)
- [7] Madden, S., Franklin, M.J.: Fjording the Stream: An Architecture for Queries over Streaming Sensor Data. In: *Proceedings of the 18th International Conference on Data Engineering*, February 2002, pp. 555–566 (2002)
- [8] Hammad, M.A., Aref, W.G., Elmagarmid, A.K.: Stream Window Join: Tracking Moving Objects in Sensor-Network Databases. In: *Proceedings of the 15th International Conference on Scientific and Statistical Database Management (SSDBM 2003)* (July 2003)

- [9] Madden, S., Shah, M., Hellerstein, J., Raman, V.: Continuously adaptive continuous queries over streams. In: Proc. of the SIGMOD Conference (June 2002)
- [10] Kang, J., Naughton, J.F., Viglas, S.D.: Evaluating window joins over unbounded streams. In: ICDE (February 2003)
- [11] Pandit, A., Gupta, H.: Communication-Efficient Implementation of Range-Joins in Sensor Networks. In: Li Lee, M., Tan, K.-L., Wuwongse, V. (eds.) DASFAA 2006. LNCS, vol. 3882, pp. 859–869. Springer, Heidelberg (2006)
- [12] Yu, H., Lim, E.-P., Zhang, J.: In-Network Join Processing for Sensor Networks. In: Zhou, X., Li, J., Shen, H.T., Kitsuregawa, M., Zhang, Y. (eds.) APWeb 2006. LNCS, vol. 3841, pp. 263–274. Springer, Heidelberg (2006)
- [13] Coman, A., Nascimento, M.A., Sander, J.: On Join Location in Sensor Networks. In: The 8th International Conference on Mobile Data Management, pp. 190–197 (2007)
- [14] Stern, M.: Optimal Locations for Join Processing in Sensor Networks. In: The 8th International Conference on Mobile Data Management, pp. 336–340 (2007)
- [15] Coman, A., Nascimento, M.A.: A Distributed Algorithm for Joins in Sensor Networks. In: 19th International Conference on Scientific and Statistical Database Management (2007)
- [16] Gupta, H., Chowdhary, V.: Communication-efficient Implementation of Join in Sensor Networks. *Ad Hoc Networks* 5(6), 929–942 (2007)
- [17] Kwan, E., Getta, J.R., Vossough, E.: Design and Implementation of Data Stream Processing Applications. In: 2nd International Conference on Software and Data Technologies (2007)
- [18] Vossough, E.: A System for processing continuous Queries over infinite data streams. In: Galindo, F., Takizawa, M., Traunmüller, R. (eds.) DEXA 2004. LNCS, vol. 3180, pp. 720–729. Springer, Heidelberg (2004) ISSN: 0302-9743
- [19] Vossough, E.: Processing Of Continuous Queries Over Infinite Data Streams. VDM Verlag Dr. Mueller e.K (2008) ISBN-13: 9783639045475

Facilitating Reuse of Code Checking Rules in Static Code Analysis

Vladimir A. Shekhovtsov, Yuriy Tomilko, and Mikhail D. Godlevskiy

Department of Computer-Aided Management Systems,
National Technical University “Kharkiv Polytechnical Institute”, Ukraine
shekvl@yahoo.com, tomilko.yuriy@mail.ru, god_asu@kpi.kharkov.ua

Abstract. Currently, the rationale of applying code checking rules in static code analysis is often not captured explicitly which leads to the problems of rule reuse in similar development contexts. In this paper, we investigate the process of tracing possible sources of such rules back to design decisions and quality requirements. We present an idea of storing the rationale information along with particular code checking rules in a rule repository. We argue that such information is related to particular design decisions or patterns that need to be enforced by the rule and to generic properties of these decisions such as corresponding quality characteristics. We show how a reuse support tool with underlying rule repository can aid in defining the recommended set of rules to be reused while making recurring design decisions or applying design patterns.

1 Introduction

Static code analysis is a special kind of inspection [17] which is performed over the source (or bytecode-compiled) code with a goal of revealing and correcting errors introduced during software development [1, 33]. It also helps assessing software quality attributes and checking standards compliance. Specific static analysis tools [29, 31] (such as FxCop, Gendarme, NDepend, FindBugs, PMD etc.) allow the user to apply the set of *code checking rules* to the code looking for known bugs, various violations of common recommendations and policies, other code deficiencies. Applied properly, static code analysis tools free the code analyst from routine work while allowing concentrating at more complex tasks.

Working with these tools, however, presents its own set of issues. Producing the full set of code checking rules for the project requires significant effort. Predefined sets of rules packaged with tools seldom satisfy the developer; in these cases custom rules need to be created. If such rules have to be implemented using general-purpose languages such as Java or C# [5, 6, 13], this becomes time-consuming and routine work: it is necessary to provide a technical “boilerplate” code enabling the features provided by the tool (e.g. initializing its API), implement the rule-checking code itself, and create the set of configuration parameters allowing run-time rule customization. As a result, *rule reuse problem* becomes important. Unfortunately, reusing complete sets of rules is not an option in most cases as the projects and tools usually

differ enough to make such action impossible. On the other hand, it is desirable to perform at least partial rule reuse as the projects often require similar rules.

One of the problems with rule reuse is that currently the rationale of applying the rules in particular projects is often not documented explicitly. Rules appear “out of developer’s experience” as no rule traceability back to the previous stages of the software process is supported. As a result, the reuse of such rules becomes difficult as nothing suggests the developer that the current project offers the prerequisites for the rules already used in some previous project.

In this paper, we will show how the rule reuse problem can be addressed via investigating the rationale behind the code checking rules. To do this, we turn to earlier steps of the software process (architectural design and requirements engineering) and look at design decisions, design patterns, and software quality requirements as possible candidates for this rationale. We argue that knowing the origin of every rule helps to achieve its reuse in future if the same context is encountered again during the development. To facilitate reuse, we propose establishing the rules repository where the rationale information is stored alongside the information about the rules themselves.

The rest of the paper is organized as follows. Section 2 presents the background information. Section 3 investigates possible rationale for application of the rules; section 4 presents a conceptual model of code checking rules and their rationale which can be used as a foundation of rules repository schema. Section 5 outlines possible reuse cases and scenarios using the knowledge of the code checking rules rationale. Section 6 surveys the related work, following by a conclusion and future work description in Section 7.

2 Background

In this section, we briefly introduce code checking rules and some candidates for their rationale, in particular, design decisions and quality requirements.

2.1 Code Checking Rules

The main purpose of static code analysis tools [29, 31] is establishing the set of code checking rules looking for common bugs and bad practices in software and reporting the results to the analyst. Such rules define undesirable or prohibited code solutions (code anti-patterns) usually expressed in terms of the constructs of the particular target programming language (class, interface, method, property etc.) and other applicable (e.g. API-related) concepts. Such solutions include e.g. invalid usage of a particular class, undesirable sequence of method calls, improper access to a particular method from other particular method, violating some guidelines for class implementation or API usage (e.g. failing to implement a mandatory interface or property).

Let us look at an example of a code checking rule. Maintaining dependencies between the components in multi-tier architecture may lead to the restriction of using particular set of classes from other particular set of classes, for example, presentation tier classes cannot be used from business logic classes. This leads to the following rule: *it is forbidden to use types belonging to a presentation tier from any types belonging to*

a *business logic tier*. An example of an implementation of this rule (using Code Query Language (CQL) [3] supported by NDepend tool) is as follows:

```
WARN IF Count > 0 IN SELECT METHODS FROM "BusinessLogic"
WHERE IsDirectlyUsing "PresentationLogic"
```

2.2 Design Decisions and Quality Requirements

Design decisions are “the significant decisions about the organization of a software system” [22]. Such decisions are made during the design step of the software process and reflect the view of the system that needs to be implemented. Actually, every design decision directly controls the code development. The extensive ontology of such decisions is proposed by Kruchten [23], it identifies three major decision categories: existence decisions (related to some functionality present or absent in a system), property decisions (related to some quality attribute supported by a system) and executive decisions (not related directly to the functional elements or their qualities). In recent works, the complete software architecture tends to be viewed as a set of such decisions [20]. Examples of design decisions can be “the system will have three-tier architecture”, “the system will include the cache manager using up to 5 background threads and allowing up to 1000 concurrent clients”.

Software quality requirements represent the desirable quality characteristics for a system [9, 12] (such as availability, performance, safety, or security) as seen by its stakeholders. Examples of quality requirements are “response time for operations with a customer order must not exceed 0.2 sec under the load up to 700 requests per second”, “all attempts to have access to the order processing system must be authorized”. Such requirements are actually supported or opposed by corresponding design decisions [35], we will look at this issue in detail in subsection 3.3.

3 The Rationale Behind the Code Checking Rules

In this section, we show how the rationale behind the code checking rules can be revealed. We trace this rationale first to existing design decisions and then to corresponding software quality requirements.

3.1 Code Checking Rules and Design Decisions

To find the rationale behind the application of a particular code checking rule, we need to look at the activities belonging to earlier steps of the software process. We argue that *as the set of code checking rules enforces a particular design decision, it is natural to accept that decision as a direct rationale for these rules*.

For example, the design decision “the logical view is organized in 3 tiers named DataAccessLogic, BusinessLogic and PresentationLogic” is a direct rationale for establishing the following code checking rule “no methods of classes belonging to PresentationLogic package can be called from either DataAccessLogic or BusinessLogic packages” as the creation of this rule is a direct consequence of accepting the design decision.

As a software architecture can be seen as a set of interconnected design decisions [20] and Kruchten’s design decision ontology [23] adds to this set the decisions not related to architecture (introducing the executive decisions related to organizational issues, the choice of technology or a development tool) we argue that *every code checking rule can be traced to the corresponding design decision*. Actually, such rule is an integral part of the implementation of this design decision.

The design decision usually contains some information that can be used to parameterize the corresponding rules (Fig.1). For example, if it was defined that the presentation layer code is contained in a package named *PresentationLogic*, the name of the package parameterizes the rule restricting an access to this package. Numeric information (such as cache size, maximum number of threads etc) can be used this way as well. Reusing such rules requires defining the values for specified parameters.

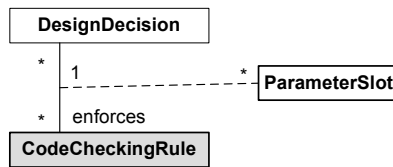


Fig. 1. Design Decision and Code Checking Rule

Specifications of design decisions can be treated to some degree as *code quality requirement specifications* as they define the desired quality properties of the corresponding implementation code. These specifications are also similar to requirement specifications as they are often informal (actually, the design decisions are often left expressed in natural language). In fact, some kind of NLP-based requirement elicitation and analysis can be performed to actually map these specifications into code checking rules (similarly to what is proposed by conceptual predesign [26] and the NIBA project [10] to map traditional software requirements into design-time notions). This mapping is a target for future research, in this paper we treat the design decisions as “black boxes” assuming they are specified in a format that allows retrieving the information intended for rule parameterization.

3.2 Code Checking Rules and Design Patterns

From the point of view of facilitating reuse, it would be even more interesting and useful to look at *design patterns* as possible rationales for code checking rules. The simplest way to deal with this case is to treat the pattern as a special kind of design decision which has predefined meaning and can be applied in different contexts (this is the view accepted in e.g. [20]). In this case, the set of code checking rules can be seen as enforcing the design pattern the same way as it enforces any other design decision. Even in this case, however, we would recommend distinguishing design patterns from other (ad-hoc) design decisions as it would greatly help in documenting the likely chances of reuse (as design patterns in most cases become reused more often than ad-hoc design decisions).

Another approach is to view design patterns as “enablers” for several design decisions at once (as original GoF book states “once you know the pattern, a lot of design decisions follow automatically” [11]). In this situation, a design pattern can be treated as a rationale for all code checking rules corresponding to the enabled decisions. For example, applying *Model-View-Controller* design pattern enables the set of existence design decisions related to defining the program units implementing all parts of a solution (such as “the package *DataView* will contain the code for the view class and all supporting program artifacts”). These decisions, in turn, are enforced by a set of code checking rules (e.g. prohibiting direct access from the model to the view).

Making able to relate code checking rules to design patterns is probably the most important reuse facilitation technique as such rules become reused frequently.

3.3 Code Checking Rules and Quality Requirements

Now we can investigate the possibility of tracing the rationale of code checking rules further – to *software quality requirements* (those defined by the stakeholders in a process of requirements elicitation). Being able to relate static code checking rules to software quality requirements allows for better understanding the interdependencies between external and internal software quality (as quality requirements represent external quality and code checking rules are supposed to enforce internal quality).

To understand the relationships between quality requirements and code checking rules, first look at the relationships between quality requirements and design decisions [35]. These relationships are depicted on Fig.2. They are of “many-to-many” kind as achieving the particular quality requirement can be affected by several design decisions (e.g. the desired performance can be achieved with a help of creating a cache with a specific size and going multithread) whereas a particular design decision can affect achieving several quality requirements (e.g. the cache of the particular size can help both performance and scalability). Also it is necessary to introduce the degrees of affection as it can be either positive or negative. Some decisions can actually break quality requirements as it is not always possible to satisfy all such requirements due to implementation constraints and some compromise need to be found ([2] uses the term “satisficing” referring to such requirements). For example, the design decision “a cache database allowing up to 500 simultaneous users will be used by a cache manager” can support the quality requirement “a response time under the load up to 500 users must not exceed 0.2 sec” while breaking the requirement “a response time under the load up to 1500 users must not exceed 0.5 sec”.

Now it is possible to trace the code checking rules to the software quality requirements. As the specific quality requirement is affected by the specific design decision either positively or negatively, it can be also affected the same way by all code checking rules that enforce this design decision. So actually we have several groups of code checking rules related to particular quality requirement – one group per degree of possible affection. In this paper we restrict ourselves to only two groups: *supporting rules* and *breaking rules*. It is important to understand, however, that if the particular rule is listed for the particular requirement as “breaking” it does not necessarily mean that its

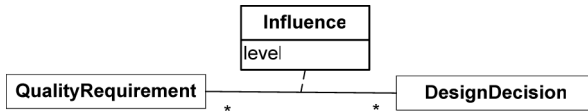


Fig. 2. Quality Requirement and Design Decision

implementation and enforcement necessary breaks this requirement: it only means that it enforces breaking design decision so it has more chances to be harmful and needs to be investigated more carefully.

Some code checking rules cannot be traced to quality requirements, in particular, the rules driven by business environment i.e. related to accepted coding standards, tools to be used etc. (corresponding to the executive decisions according to Kruchten's ontology [23]). This fact has to be captured in a way that makes it able for an analyst to locate all the rules of this kind.

3.4 Code Checking Rules and Software Quality Characteristics

Software quality requirements refer to specific quality characteristics belonging to particular quality model (such as e.g. defined by ISO/IEC 9126 quality model standard [18]). These models are usually defined as taxonomies of quality characteristics with specific software metrics belonging to the lowest level. For example performance requirement "The response time for order processing must not exceed 0.5 sec" (*R1*) refers to "response time" quality metric which belongs to "time behavior" middle-level characteristic and "efficiency" upper-level characteristic according to ISO/IEC 9126 quality model specification.

As we have already connected code checking rules to software quality requirements, we can see that this connection contains information related to association between the rules and corresponding quality characteristics. For example, all the rules related to *R1* requirement can be associated with a "response time" metric and two upper-level quality characteristics: "time behavior" and "efficiency".

Connecting code checking rules to quality characteristics can be seen as rule categorization, it does not involve parameterization of the rules (as in case of design decisions) and does not include the degree of affection (there are no "rules breaking response time" as "response time" in this case is only a category label). If such parameterization is undesirable, the analyst can even skip the detailed description of the relationship between the rule and the design decision and instead only record the connection between the rule and the corresponding quality characteristic.

Another quality-based code checking rule classification is proposed in [27]. Using the tool proposed in that paper, it is possible to specify the code quality characteristic and obtain all the rules associated with it. This approach, however, is restricted to code quality characteristics that characterize internal software quality and can be directly associated with the rules. Our approach allows revealing the indirect connection between code checking rules and external software quality characteristics. These two classifications are actually complimentary and can be used alongside each other.

4 A Conceptual Model of Code Checking Rules and Their Rationale

After achieving understanding of the relations between the rules, design decisions, and quality requirements we can describe how this information can be used by the tool aimed at facilitating rule reuse. We propose to establish the rules repository where the information about the available rules together with their rationale is stored and make the tool access this repository. The conceptual model of code checking rules making use of these relations can be used to establish a schema for this repository. This model is shown on Fig.3; for brevity, we omitted most attributes and the software project-related part of the model.

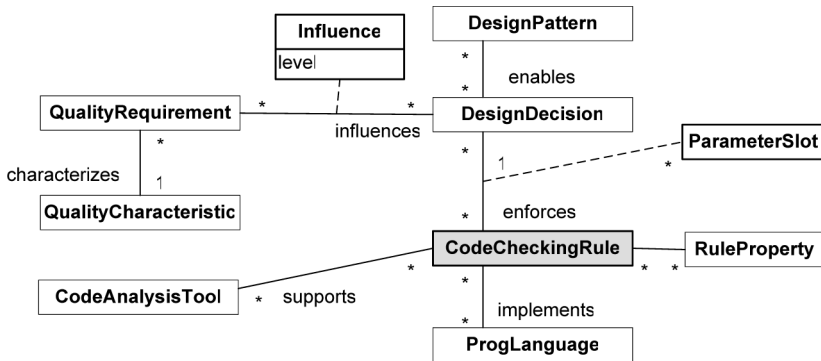


Fig. 3. A Conceptual Model of Code Checking Rules and Their Rationale

In this model, the relationships between the rules and their properties are of “many-to-many” kind. It reveals the fact that we treat these properties as “tags”; as a result, every rule can have sets of tags of different kind associated with it. This tagging approach allows for additional flexibility of expressing the rules; the repository based on this model is supposed to offer better search capabilities.

Design decisions are also treated as tags as there can be many design decisions connected to the particular code checking rule (probably originated from different projects). The model allows the analyst to drill down the particular design decision to see all the corresponding quality requirements, quality characteristics etc.

The model reflects the fact that not every programming language or static analysis tool offer the support for the particular rule (see [34] for more formal treatment of this issue). Some rules are specific for a particular language or framework (e.g. namespace-related rules for .NET); some tools, on the other hand, do not offer capabilities to implement specific categories of rules. To deal with this fact, the information about the available language and tool support is stored with the rule. If the particular support is missing, the analyst attempting to locate the rule in a repository should receive the meaningful explanation of the problem. This approach also have an advantage of being flexible enough to incorporate future events when new language constructs are

introduced to match the capabilities of other languages (e.g. Java generics) or the tools start to support new features.

The model offers limited support for rule parameterization; actually it allows defining the set of parameter slots to transfer the information between the design decision specification and the code checking rule (both treated as “black boxes” as stated earlier). How this information can be used to actually parameterize the rule is left to the implementation of the rule reuse support tool. In future, we plan to make use of language- and tool-independent metamodel (similar to what is proposed in [34]) to describe the rule functionality, in this case, we will precisely define the support for rule parameterization (treating the rule as a “white box”).

5 Rule Reuse

In this section, we look how the proposed notions of code checking rule rationale can be used in practice to facilitate reuse. We start from the common list of reuse cases and later introduce two possible rule reuse scenarios: top-down and bottom-up.

5.1 Rule Reuse Cases

Our model facilitates the reuse of the code checking rules for the following cases (suppose we have established the special repository where the information about the available rules and their rationale is registered; the structure of this information follows our conceptual model):

1. Implementing the registered design decision: it is possible to reuse all the corresponding rules directly with all the parameter values (rare case of “copy-paste” reuse), perform the corresponding parameterization, reject some rules, or use some combination of these actions.
2. Applying the registered design pattern: the most frequent action is to reuse all the corresponding rules with parameterization.
3. Encountering the registered quality requirement: it is possible to consider reusing the corresponding supporting rules (the actual reuse will be probably postponed until the actual design decisions are defined).
4. Investigating the consequences of supporting the particular registered quality characteristic: all the code checking rules corresponding to this characteristic can be considered for reuse and investigated in detail.
5. Working on a project which is similar to already-performed project: all the design decisions of the past project can be examined together with their corresponding code checking rules.

5.2 Rule Reuse Scenarios

Let us look at two typical scenarios for the reuse of the code checking rules: bottom-up and top-down.

Top-down scenario is a straightforward one:

1. The analyst implements the design decision to support the particular quality requirement. This decision has to be enforced with a set of code checking rules.

2. The search for a quality requirement in a repository is performed. If it is not found new requirement is registered, otherwise case 3 (see subsection 5.1) is followed;
3. The search for a design decision in a repository is performed. If it is not found new decision is registered in a repository, otherwise case 1 is followed;
4. If on any of the previous stages the corresponding checking rules cannot be found, they should be created and registered as well.

Bottom-up scenario should occur less frequently and is more difficult to follow:

1. The analyst works with some static code analysis tool on a particular application (it can be an existing application being refactored etc.)
2. The set of errors and imperfect code fragments is found; some of them are believed to have recurring character.
3. The synthesis of the code checking rules is performed for these errors; the information about these rules is registered in a repository.
4. These rules are associated to the possible design decisions that could serve as their rationale. This activity can involve non-trivial reasoning (in reverse engineering sense) if design decisions have yet to be defined. In complicated cases, only the categorization of the rules according to quality characteristics can be performed (this is usually easier to do).
5. If quality requirements specification is defined for a system, design decisions can be related to quality requirements.

Described reuse cases and scenarios can serve as parts of requirements specifications for a reuse support tool with underlying code checking rules repository. This tool is currently under development.

6 Related Work

We can classify the related work on rule reuse into two categories: (1) low-level approaches addressing rule reuse on the code level (“white-box” reuse) and (2) higher-level approaches investigating the place of such rules in a software process, their relations to the goals of the project, its quality requirements etc.; these approaches treat the rules as “black boxes” to some degree (with possible parameterization). Most of the work in this area falls into the first category. We found very few works investigating the application rationale of the rules.

6.1 Source-Level (White-Box) Rule Reuse

Custom code analysis rules have yet to be extensively investigated by the research community, most of the publications related to such rules are “how-to” (mostly on-line) guides targeting particular industrial [5-7, 13] or research [16] tools; few attempts to achieve reuse across tool or language boundaries are made. As a result, the proposed solutions are actually rather low-level; they concentrate on specifics of rule implementation for particular tools. An exception is a paper [7] where in addition to targeting a Fortify tool an attempt is made to express the rules using high level (language- and tool-agnostic) description language with defined grammar; no grammar, however, is actually introduced in a paper.

A common white-box approach for rule reuse is to start from building a conceptual rule model and then instantiate this model in different contexts. Jackson and Rinard in their roadmap for software analysis [19] emphasized the importance of this approach. They suggested using special generic code representation as a model and build all the code analysis activities on top of this model. The list of code models suitable to be used in model-driven software analysis is rather extensive; it includes the schemas for XML-based source code exchange formats such as GXL [15], srcML [4, 24], or OOML [25], semantic code models such as ASG [8]. Detailed discussion related to interrelationships between code models and code exchange formats is presented in [21]. The specific case of using and adapting srcML for the support of code analysis is discussed in [24].

An extensible meta-model-based framework to support the code analysis is described in [34]. This approach provides the most extensive white-box reuse capabilities. It formally defines the relationships between generic representation of the source code, conditions specified on top of this representation (which can represent code analysis rules), and specifics of particular languages and tools (front-ends for the common metamodel). The definition of the common meta-model underlying the framework reflects these issues, rules expressed via this metamodel can be easily reused across tool and language boundaries. The metamodel, however, does not provide any means to aid capturing design-time rationale of rule application.

6.2 Code Quality and Black-Box Rule Reuse

An approach utilizing code quality models to assess open-source software projects using extensive set of metrics was developed as a result of an SQO-OSS project [30]. Other approaches aimed at this goal are presented in [28, 32]. Such approaches are limited to integrated quality assessment (i.e. they produce the values for software quality attributes and an integrated value for overall product quality), they do not address rules necessary to enforce quality. Their quality characteristics, however, can be useful in reuse contexts as the rules can be associated with such characteristics using an approach shown in subsection 3.4.

Connecting code checking rules to a quality model was proposed in [14, 27]; we briefly discussed this work in the subsection 3.4. This connection is introduced as a part of specific code analysis process called EMISQ. Both quality model and an evaluation process are based on ISO 14598 and ISO 9126 standards. No model for internal structure of the rules is proposed, instead, the available rules are stored into the repository after classifying according to (1) code quality characteristics they help to enforce and (2) available implementations (language and static analysis tool support). The proposed tool aids the developer allowing fast filtering of the available set of rules according to a quality characteristic they are supposed to enforce, required tool or language support etc. The EMISQ approach is close to our technique with respect to admitting the need for capturing the connections from the rules back to the system goals (which can be seen [2] as corresponding to software quality requirements). Actual implementation of this connection, however, is not presented in a paper and its use to facilitate rule reuse is not investigated. Also this approach does not take into account design decisions and patterns. In fact, to aid rule reuse, the developer is supposed to pick context-relevant code quality characteristic manually and choose the appropriate rules without any specific connection to the past development experience.

7 Conclusions and Future Work

In this paper, we proposed an approach for revealing the rationale behind the application of code checking rules in static code analysis. We stated that these rules can be related to design decisions and patterns, quality requirements and quality characteristics. We presented a conceptual model for the code checking rules and their rationale; this model can be used to establish a schema for the rule repository underlying reuse support tools. The proposed solution facilitates rule reuse in recurring development contexts by allowing the analyst to look at all the rules related to the particular context.

In future, we plan to implement a tool support for our approach. Prospective tool will offer a repository for code checking rules organized in correspondence with the proposed conceptual model allowing analysts to perform queries for available rules according to different reuse cases and scenarios. Another direction of research is transition to the “white box” representations of the code checking rules (according to [34]) and the design decisions (e.g. according to their UML profile [35]).

References

1. Chess, B., West, J.: *Secure Programming with Static Analysis*. Addison-Wesley, Reading (2007)
2. Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J.: *Non-Functional Requirements in Software Engineering*. Kluwer Academic Publishers, Dordrecht (1999)
3. Code Query Language 1.8 Specification (accessed January 11, 2008), <http://www.ndepend.com/CQL.htm>
4. Collard, M.L., Maletic, J.I., Marcus, A.: Supporting Document and Data Views of Source Code. In: *Proc. DocEng 2002*. ACM Press, New York (2002)
5. Copeland, T.: Custom PMD Rules. OnJava.com (2003) (accessed January 11, 2008), http://www.onjava.com/pub/a/onjava/2003/04/09/pmd_rules.html
6. Create Custom FxCop Rules (accessed January 11, 2008), <http://www.thescarms.com/dotnet/fxcop1.aspx>
7. Dalci, E., Steven, J.: A Framework for Creating Custom Rules for Static Analysis Tools. In: *Proc. Static Analysis Summit*, pp. 49–54. Information Technology Laboratory, NIST (2006)
8. DATRIX Abstract Semantic Graph Reference Manual, version 1.4. Bell Canada (2000)
9. Firesmith, D.: Using Quality Models to Engineer Quality Requirements. *Journal of Object Technology* 2, 67–75 (2003)
10. Fliedl, G., Kop, C., Mayerthaler, W., Mayr, H.C., Winkler, C.: The NIBA Approach to Quantity Settings and Conceptual Predesign. In: *Proc. NLDB 2001*. LNI, vol. P-3, pp. 211–214. GI (2002)
11. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: *Design patterns. Elements of reusable object-oriented software*. Addison-Wesley, Reading (1995)
12. Glinz, M.: Rethinking the Notion of Non-Functional Requirements. In: *Proc. Third World Congress for Software Quality (3WCSQ 2005)*, Munich, vol. II, pp. 55–64 (2005)
13. Grindstaff, C.: FindBugs, Part 2: Writing custom detectors. IBM Developer Works (2004) (accessed January 11, 2008), <http://www.ibm.com/developerworks/library/j-findbug2>

14. Gruber, H., Körner, C., Plösch, R., Schiffer, S.: Tool Support for ISO 14598 based code quality assessments. In: Proc. QUATIC 2007. IEEE CS Press, Los Alamitos (2007)
15. Holt, R.C., Winter, A., Schürr, A.: GXL: Toward a Standard Exchange Format. In: Proc. WCRE 2000, pp. 162–171 (2000)
16. Holzmann, G.J.: Static Source Code Checking for User-Defined Properties. In: Proc. IDPT 2002. Society for Design and Process Science (2002)
17. IEEE Standard for Software Reviews. IEEE Std 1028-1997. IEEE (1997)
18. ISO/IEC 9126-1, Software Engineering – Product Quality – Part 1:Quality model. ISO (2001)
19. Jackson, D., Rinard, M.: Software Analysis: A Roadmap. In: Proc. Conf. on The future of Software engineering. ACM Press, New York (2000)
20. Jansen, A., Bosch, J.: Software Architecture as a Set of Architectural Design Decisions. In: Proc. WICSA 2005, pp. 109–120. IEEE CS Press, Los Alamitos (2005)
21. Jin, D.: Exchange of software representations among reverse engineering tools. Technical Report. Department of Computing and Information Science, Queen's University, Kingston, Canada (2001)
22. Kruchten, P.: The Rational Unified Process - An Introduction. Addison-Wesley, Reading (1995)
23. Kruchten, P.: An Ontology of Architectural Design Decisions in Software-Intensive Systems. In: 2nd Groningen Workshop on Software Variability Management (2004)
24. Maletic, J.I., Collard, M.L., Kagdi, H.: Leveraging XML Technologies in Developing Program Analysis Tools. In: Proc. ACSE 2004, pp. 80–85. The IEE Publishers (2004)
25. Mamas, E., Kontogiannis, K.: Towards Portable Source Code Representations Using XML. In: Proc. WCRE 2000, pp. 172–182. IEEE CS Press, Los Alamitos (2000)
26. Mayr, H.C., Kop, C.: Conceptual Predesign - Bridging the Gap between Requirements and Conceptual Design. In: Proc. ICRE 1998, pp. 90–100. IEEE CS Press, Los Alamitos (1998)
27. Plösch, R., Gruber, H., Hentschel, A., Körner, C., Pomberger, G., Schiffer, S., Saft, M., Storck, S.: The EMISQ Method - Expert Based Evaluation of Internal Software Quality. In: Proc. 3rd IEEE Systems and Software Week. IEEE CS Press, Los Alamitos (2007)
28. Rentrop, J.: Software Metrics as Benchmarks for Source Code Quality of Software Systems. Vrije Universiteit, Amsterdam (2006)
29. Rutar, N., Almazan, C.B., Foster, J.S.: A Comparison of Bug Finding Tools for Java. In: Proc. ISSRE 2004, pp. 245–256. IEEE CS Press, Los Alamitos (2004)
30. Samoladas, I., Gousios, G., Spinellis, D., Stamelos, I.: The SQO-OSS quality model: measurement based open source software evaluation. In: Proc. OSS 2008, pp. 237–248 (2008)
31. Spinellis, D.: Bug Busters. IEEE Software 23, 92–93 (2006)
32. Stamelos, I., Angelis, L., Oikonomou, A., Bleris, G.L.: Code quality analysis in open source software development. Info. Systems J. 12, 43–60 (2002)
33. Stellman, A., Greene, J.: Applied Software Project Management. O'Reilly, Sebastopol (2005)
34. Strein, D., Lincke, R., Lundberg, J., Löwe, W.: An Extensible Meta-Model for Program Analysis. IEEE Transactions on Software Engineering 33, 592–607 (2007)
35. Zhu, L., Gorton, I.: UML Profiles for Design Decisions and Non-Functional Requirements. In: Proc. SHARK 2007. IEEE CS Press, Los Alamitos (2007)

Achieving Adaptivity Through Strategies in a Distributed Software Architecture

Claudia Raibulet¹, Luigi Ubezio², and William Gobbo²

¹ Università degli Studi di Milano-Bicocca, DISCo – Dipartimento di Informatica Sistemistica e Comunicazione, Viale Sarca 336, Edificio 14, 20126 Milan, Italy
raibulet@disco.unimib.it

² IT Independent Consultant,
Milan, Italy
ubezio@gmail.com, william.gobbo@hotmail.it

Abstract. Designing information systems which are able to modify their structure and behavior at runtime is a challenging task. This is due to various reasons mostly related to questions such as what should be changed, when should be changed, and how should be changed at runtime in order to maintain the functionalities of a system and, in the same time, to personalize these functionalities to the current user, services requests and situations, as well as to improve its performances. The systems which manage to address properly these aspects are considered adaptive. Our approach to design adaptive systems exploits strategies to implement the decisional support and to ensure an efficient modularity, reusability and evolvability of the architectural model. In this paper we describe the main types of the strategies defined in our solution, as well as how these strategies are exploited at run-time in the context of an actual case study in the financial domain.

1 Introduction

Adaptivity is one of the keywords related to the design of today's information systems. It addresses the modifications performed in a system during its execution. These modifications aim to improve the productivity and performance of a system, and to automate the configuration, re-configuration, control and management tasks. They may be translated into modifications of structural, behavioral or architectural components [5, 6, 8].

This paper presents our solution for the design of an Adaptive MANagement of Resources with Strategies (ARMANIS) in a service-oriented mobile-enabled distributed system. In this context, we focus on the definition of various types of strategies, which implement the decisional support playing a fundamental role in the process of achieving adaptivity. They exploit a late binding mechanism which enables us to combine structural and behavioral elements in order to obtain a personalized solution for each request based on its input information.

In previous work, the architectural model of our solution which exploits reflection to achieve adaptivity has been described [10]. In the same paper we have presented an

implementation example based on a distributed peer-to-peer paradigm. The service-oriented features of ARMANIS have been described in [12]. We have validated the proposed model by applying it in a healthcare system [11]. Furthermore, we have adapted and implemented our solution for mobile devices [2]. In this paper, we aim to focus the attention on the strategies we consider to be fundamental to achieve adaptivity through ARMANIS. In [10, 14] we have introduced the concept of strategy and described its design and implementation issues through the Strategy and Composite design patterns. In this paper we describe the types of strategies to be considered and exploited at runtime by the various components of our architectural model: domain, system and connectivity. The definition of these strategies can be considered a further evolution of our approach, evolution mostly due to the integration of the wired (e.g., LAN) and wireless (e.g., Bluetooth, WI-FI) networks, the usage of the RFID (Radio Frequency Identification) [4, 15] technology for the localization of the system actors, and to the possibility to apply it in various application contexts.

The rest of the paper is organized as following. Section 2 introduces the case study considered to validate our solution. Section 3 describes our architectural model by focusing on its main concepts meaningful for this paper. The application of our approach in the context of the case study is dealt in Section 4. Discussions and further work are presented in Section 5.

2 Motivating Example

The motivating example is collocated in the context of a finance case study. The main idea behind this example is to provide support to the bank staff members to offer customers entering a bank agency personalized services based on customers' profile and account status, as well as on the current business and financial advertisings offered by the bank. Customers appreciate and agree easier to exploit new and personalized services when the bank staff members have the chance to explain them face to face the advantages of the advertisings, rather than when the bank contacts them through communication letters, emails or phone calls.

A prerequisite to achieve this goal is to reveal the presence of the customers entering the bank agency in a non-intrusive way. A possible scenario may be based on a RFID approach, which notifies the system whenever a person having a tag crosses an access point. We suppose that the bank customers have a credit card enriched with a tag, which is able to be excited by radio signals and to send back its own identifier. We have already developed a complex access control framework for the management of multi-services structured based on the notification of the customers' presence [16]. In this case study the access point, called choke point, is the agency entrance(s).

A general scenario consists in the following steps. The system reveals the presence of the customers through their enhanced credit cards. Furthermore, the system looks for the customer's account and identifies the services he can be interested in. Based on the commercial importance of the customer for the bank, various staff members are notified of his presence (e.g., the cashiers, an account consultant, the director of the agency). This information may be displayed on various devices used by the bank staff members (e.g., desktop monitors, wall monitors, PDAs, mobile phones) and in various modes based on their current location and the activities they are performing.

We underline that the actual users of this system are the staff members of the bank agency. The system provides them support to experience a new way of interacting with the bank customers.

Related to this case study we present the following two significant scenarios.

Scenario 1: The bank staff members are notified when a customer enters a bank through a message on their working device. In this scenario, we consider an ordinary customer who interacts only with one of the cashiers at work.

Through the credit card, the system reveals the presence of the customer and identifies the customer's name and bank account number. Based on the customer's profile, the system identifies if there are any advertisements which should be communicated to the customer. All this information is displayed on the desktops used by the cashiers.

Scenario 2: An important customer enters the bank. In this case the director of the bank agency is notified. The way he is notified depends on his location (e.g., office room, meeting room, out of the bank agency) and of the device he has available (e.g., working terminal, wall monitor, PDA, mobile phone). If the director is not present in the agency, the vice-director is notified.

3 ARMANIS' Architectural Model

Figure 1 shows the main components of our architectural model:

- the Domain Manager, which provides a Graphical User Interface (GUI), a Request Execution Protocol (REP) module, and the knowledge specific to the current application domain; the GUI shows to the users the domain services offered by the software system; the REP module defines the steps for the execution of services' requests in the context of the application domain;
- the System Manager, which controls the services provided by the hardware and software components of a system (e.g., print, display); these services are independent of the application domain and depend only on the system's architecture; system services are exploited by the domain services;
- the Connectivity Manager, which supervises the communication with other ARMANIS-enabled nodes; this manager deals with various types of networks both wired (e.g., LAN) and wireless (e.g., WI-FI, Bluetooth).

As shown in Figure 1, the three managers are related to the three main types of knowledge available in every software system. We consider them separately to ensure the modularity of the solution, its maintenance, as well as the reusability as much as possible of the design and implementation components. Each of these managers consists of three main elements: knowledge, services and strategies. The knowledge represents the information owned by each manager. The services represent the functionalities they offer to the users or other components of a system. The strategies implement the mechanisms through which runtime adaptation is achieved.

In the remaining of this section attention is focused on the description of those elements of our architectural model which are independent of any application domain. Hence, we describe the main concepts used by the Request Execution Protocol module to indicate how services are executed. Furthermore, the strategies defined by the system and connectivity modules are introduced.

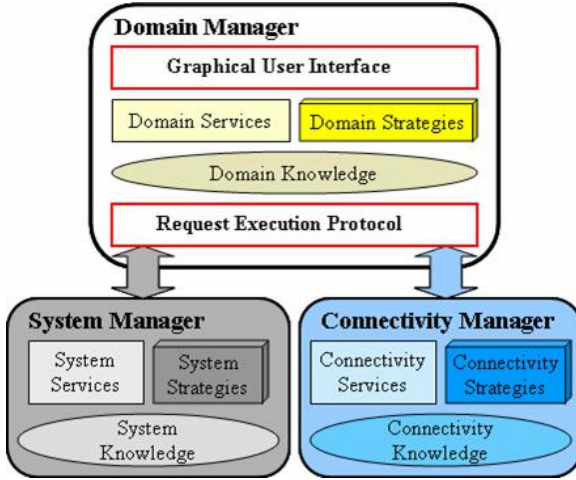


Fig. 1. ARMANIS' Architectural Model

3.1 The Request Execution Protocol Model

The Request Execution Protocol module describes formally the behavior of our approach related to the execution of services requests. The main elements of this module are shown in Figure 2. A *ServiceRequestType* defines the *States* of a request and the *Transitions* from one state to another. Each *ServiceRequest* instance corresponds to a *ServiceRequestType*. For each service request, the Checker element verifies what type it belongs to and assigns it the identified type. If there is no type matching, the request cannot be addressed.

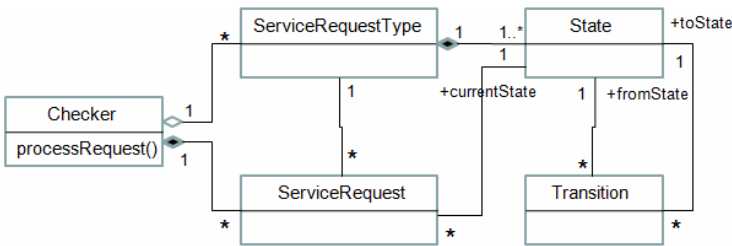


Fig. 2. Elements of the Request Execution Protocol

The definition of service execution protocols is based on:

- an XML description of the requests;
- a sequence of steps expressed in terms of a finite state machine chart; this sequence guarantees the order of the steps to execute a request.

In ARMANIS all the information exchanged among peers is document centric.

3.2 System Strategies

The System Manager performs five main tasks: the inspection of the system services offered by the local peer, the inspection of the qualities of services (QoS) of the system services offered by the local peer, the discovery of system services on remote peers, the choice of the most appropriate service for a request (considering local and remote services) and the execution of services. Thus, it defines five types of strategies one for each of these tasks. These strategies are described in Table 1. Each strategy may have one or more alternatives.

Table 1. System Strategies

Strategy Name	Basic Version	Alternative Version(s)
Service Inspection	The services offered by a peer are established statically at the start up of the system.	1. The services offered by a peer are discovered dynamically on demand. 2. The services offered by a peer are discovered dynamically at a predefined time interval.
QoS Inspection	The QoS of each service are established statically at the start up of the system and have associated quantitative values.	1. The QoS of each service are established statically at the start up of the system and have associated quantitative or qualitative values 2. The QoS of each service are discovered dynamically on demand. 3. The QoS of each service are discovered dynamically at a predefined time interval.
Service Discovery	The discovery of services offered by other peers is performed at start up. Each interrogated peer provides only its local services.	1. The discovery of services offered by other peers is performed at start up. Each interrogated peer provides its local services and the services of the peers to which it is directly connected. A number of intermediate peers can be established in order to avoid loops. 2. The discovery of services offered by other peers is performed on demand. 3. The discovery of services offered by other peers is performed at the start up and whenever the number of reachable services is less than a given value.
Service Choice	The choice of the service regards the best one available considering both local and remote services.	1. The choice of the service regards the best local available one. 2. The choice of the service regards the best remote available one. 3. The choice of the service regards the most appropriate one based on the QoS specified in the request. 4. The choice of the service regards the nearest one which is also the most appropriate one based on the QoS specified in the request. 5. The strategy identifies a list of the most appropriate services and it is the user or another component of the system to choose the one to be used based on external criteria.
Service Execution	The execution is performed by the service(s) identified through the Service Choice strategy. If errors occur, this strategy retries one more time to execute the service.	1. The execution is performed by the service identified through the Service Choice strategy. If the execution fails or errors occur, the requester of the service is notified that the service execution failed. 2. The execution is performed by the service identified through the Service Choice strategy. If the execution fails or errors occur, the Service Choice strategy is requested to identify another service. This is transparent for the requester of the service. 3. The execution is performed by the service identified through the Service Choice strategy. If the execution fails or errors occur and the Service Choice strategy has provided a list of the most appropriate services to execute the current request, than the next service in the list is chosen. If the list ends without executing the service, the requester of the service is notified of the failure.

The service inspection strategies are exploited to identify which are the services offered by a peer. The services are described through a common ontology. The basic version of this strategy considers that services are static, meaning that no services can be added or removed at run-time. Such a strategy is applicable in most of the cases. For example, the services offered by a server or a mobile phone hardly change at runtime. However, there are cases in which new types of services are added or a new type of network is available, thus alternatives to this strategy considering this aspect have been also defined.

Similar strategies have been inserted for the identification of the QoS associated to each available service. Furthermore, QoS may have both qualitative and quantitative values. For more information about the mapping between high-level QoS and low-level QoS see [13].

The service discovery system strategies are defined to inspect the services offered by other peers in the distributed system. Two approaches can be defined for this type of strategy. Through the first, only the peers directly connected to the requester one are interrogated. The second gives the possibility to specify how many intermediate peers should be interrogated. For example, establishing the intermediary peers at 1, peer P1 sees the S1.1, S1.2, S2.1, S2.2, S3.1, and S3.2 services provided by the P1, P2 and P3 peers (see Figure 3). In this case P1 cannot reach the services provided by peer P4.

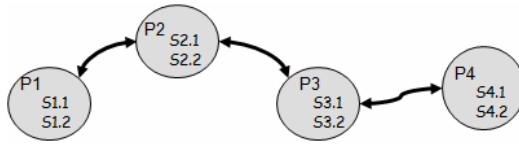


Fig. 3. Connection among peers

The service choice strategies decide which system service is used to execute the current request. In our approach, this choice is performed based on the QoS and the location of services [10]. To each service having the type of the requested one is assigned a score which indicates how close the QoS and its location are with respect to those requested. The lower the score is, the better it suits to execute the service. This strategy may provide also a list of the most suitable services which can fulfil the request. This option is very useful when the user wants to choose himself the service, or when other components of a system choose the most appropriate service. In the case of execution failures, having a two or more services which can execute the request avoids the overheads needed to identify other candidates.

3.3 Connectivity Strategies

The Connectivity Manager performs three main tasks: the discovery of other ARMANIS-enabled peers, the management of the connection among peers, and the management of the disconnection of peers. Thus, it defines three types of strategies one for each of these tasks (see Table 2).

The strategy to discover ARMANIS-enabled peers depends on the type of the network. In a highly dynamic system, the discovery of peers is done permanently, while in an almost static network discovery it is done at the start-up and/or on demand. An intermediary approach considers that the discovery of other reachable peers is done when the number of the identified or connected peers is less than a minimum number.

The opening of connections to remote ARMANIS-enabled peers depends on the type of the application domain. If it is based on a high communication and collaboration among peers, then the connection is established whenever a peer is discovered. Otherwise, it is more efficient to establish a connection on demand. Furthermore, connections may be opened when the number of the connected peers or the number of the responses received from the connected peers is less than a specific value.

Table 2. Connectivity Strategies

Strategy Name	Basic Version	Alternative Version(s)
Peer Discovery	The discovery of peers starts when the system is initialized and never ends.	1. The discovery of peers is performed at the start up of the system. 2. The discovery of peers is performed at the start up of the system and it is done whenever the number of the discovered peers is less then a given value. 3. The discovery of peers is performed at the start up of the system and it is done whenever the number of the connected peers is less then a given value. 4. The discovery of peers is performed on demand.
Peer Connection	The connection to a remote peer is performed whenever a peer is discovered.	1. The connection to remote peers is performed at the start up of the system when peers are discovered. 2. The connection to a remote peer is performed on demand and the peer is chosen among the ones identified during the start up of the system. 3. The connection to remote peers is performed at the start up of the system and it is done whenever the number of the connected peers is less then a given value. 4. The connection to remote peers is performed at the start up of the system and it is done whenever the number of the successful responses to a service request is less then a given number.
Peer Disconnection	The disconnection of a peer is performed when a communication exception is caught.	1. The disconnection of a peer is performed during the discovery process when a peer is no more reachable.

The disconnection of peers is done whenever a communication exception is caught due to the fact that a peer does not respond before a specific time limit. An alternative is to disconnect peers during the discovery process when peers become unavailable.

4 ARMANIS Applied to a Case Study

The domain knowledge for the case study introduced in Section 2 is composed of information related to customers, staff members and the financial and business services offered by the bank. Information related to the customers includes personal data, account number and conditions, and contracted services. The information of the staff members are related to their personal data, qualification and role(s), and access rights. In addition, for each staff member there are indicated alert channels which are related to the type of devices he can use for professional activities and the modality of notification. For example, a cashier uses always a desktop computer. He receives alerts of different importance in different ways: through a simple pop-up alert, a modal pop-up requiring user intervention to be closed, or a full screen message. A consultant may use a desktop or a laptop. When he uses the desktop, his location is fixed, while when using the laptop he may change his location. The director of the bank agency may use a desktop, a laptop, a PDA or a mobile phone.

The system knowledge is composed of all the devices (e.g., monitors, servers, desktops, laptops, PDAs) available in the bank agency, the services they provide (e.g., display, print, photocopy) and their related QoS (e.g., resolution, dimension, number of printed pages per minute). Each device with computational characteristics is considered an ARMANIS peer. For example, a printer is not considered a peer. It is connected to a printer server or to a desktop and the last devices are those which provide the printing service.

Due to the fact that this case study is not a very dynamic one, the service and QoS inspections are performed on demand when an upgrade is done. The service inspection is also performed on demand. The strategies related to the service choice are

different for different services. For example, in the case of choosing a printer, version number 4 (see Table 1) is adopted: the nearest one which is the most appropriate one based on the QoS specified in the request (e.g., A3 format and color printer). In the case of the display service, version number 1 is chosen: the best local available one.

For the execution of services, the basic version of the strategy is used in the context of this case study.

The connectivity knowledge is related to the topology of the bank agency network. The discovery of peers is done on demand. For example, it is performed when a staff member enters or leaves the bank with one of the devices through which he performs professional tasks. The presence detector identifies the staff member who transits an access point, and this event is the trigger for the execution of the peer discovery strategy.

The basic version of the peer connection strategy is used in this case study. For the disconnection strategy version 1 has been adopted.

Figure 4 shows the main steps performed by the system when a customer enters the bank agency. It shows the key points where strategies are used.

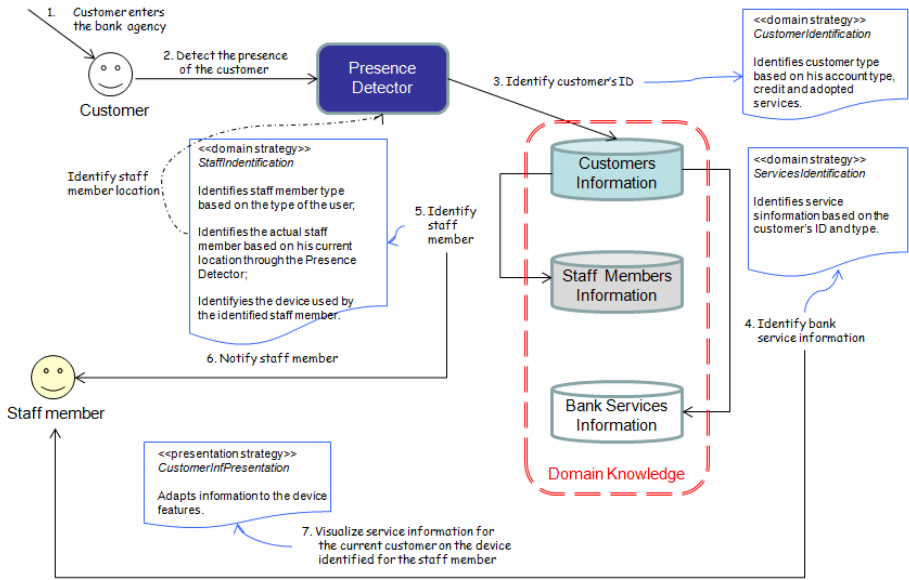


Fig. 4. The Main Steps for the Finance Case Study related to the Notification of a Customer's Presence to the Staff Members

In the following, we describe how ARMANIS is exploited in the scenarios introduced in Section 2.

Scenario 1: In this scenario only domain knowledge is used. Adaptivity is performed in the context of the domain knowledge. The system tries to extract the best matching between the customer's profile and the advertisings offered by the bank.

The customer's presence is revealed by the presence detector module: based on the credit card identification tag, the system is able to identify and access the customer's information. We have designed a domain specific strategy called *CustomerIdentification* (see Figure 4) to categorize the customer based on his account profile and exploited services. Such a strategy depends on several factors which may be independent of the account information (e.g., when the customer is an important business or political person) or dependent on the account information (e.g., the customer has a significant amount of non-invested money). This strategy may change depending on internal (e.g., the current objectives of the bank) or external factors (e.g., financial crisis) and, consequently, consider one customer as important even if previously he was considered an ordinary one, or vice-versa.

Once the customer has been identified and categorized, two further activities are performed (see Figure 4 – step 4 and 5). These activities may be fulfilled concurrently. One is related to the identification of the services which may be proposed to the customer. This step exploits a domain strategy called *ServicesIdentification* (see Figure 4). Actually, this strategy may be composed of several strategies which are combined in order to provide the best solution for the current client. They consider the available funds of the customer, the services he uses, the services he is not using yet, as well as the current advertisings of the bank. The other activity regards the identification of the alert channels related to the staff members to be notified of the presence of the customer based on the last's type. This task is performed by the domain strategy called *StaffIdentification* (see Figure 4). In this scenario, this step consists in the identification of the cashiers at work and present at their working stations.

After these two activities are fulfilled the system notifies the identified cashiers of the presence of the customer and displays the financial services extracted to be proposed to the customer in an appropriate way decided by the *CustomerInfNotification* strategy (see Figure 4).

The states of the system to perform these activities related to the notification of a customer's presence to the staff members are described in Table 3.

The staff members request further information while interacting with the customers.

At the end of the meeting, the staff members update the information related to the customers with the communications made to the customers and the obtained results in order to avoid repetitions in the future.

Table 3. Activities and states in the REP module for the notification of the customer's presence to the staff members

State Nr.	Activity Name	State Name
1.	BusyWaitPresenceNotification	CustomerPresenceNotification
2.	IdentifyCustomer	CustomerIdentification
3.	ExtractCustomerInformation	CustomerInformation
4.	IdentifyCustomerType	CustomerType
5.	IdentifyStaffMember	StaffMemberIdentification
6.	IdentifyAdvertisings	AdvertisingsIdentification
7.	NotifyStaffMember	StaffMemberNotification

Scenario 2: In the second scenario all types of knowledge are exploited. Besides the adaptivity issues related to the domain knowledge common with the first scenario, in this case adaptivity exploits also the system and connectivity parts of the system.

When an important customer enters the bank agency the director is notified. The domain manager identifies the devices registered for the director. The detection module identifies the current location of the director. Actually, this activity is performed only if the director is present in the agency, otherwise the system is already aware that he is out of the agency. This is due to the disconnection strategy of the connectivity manager which is notified by the presence detector module when the director leaves the agency and the domain specific strategy which manages the presence of the staff members at work.

If the director is in his office, the notification arrives on his desktop through a modal pop-up message. In all the other locations inside the agency, the notification arrives on the closest device (i.e., reachable peer identified by the service choice strategy of the system manager) where more information may be displayed at once.

If the director is not in the bank agency, then the *StaffIdentification* strategy (see Figure 4) requires the notification of the vice-director (or the next staff member in the hierarchy at work) and the notification of the director on his mobile phone.

4.1 Implementation Notes

The current implementation of our approach considers three types of peers: front-end, central and service. Front-end peers run applications which are used by the staff members. Applications are written in C# (for desktops) and J2ME (for mobile devices). This type of peers communicates through XML-RPC protocol with the peers storing domain information and offering services. The central peers are used to store information related to the staff members and to the customers and to offer domain specific services. In addition, they store information about the passage of customers through choke points (e.g., customers entering or leaving a bank agency). Applications running on central peers are written in the Java language due to its portability feature which allows them to be independent of any operating system. The service peers process signal information revealing the presence of the customers, which are further sent and stored on the central peers. Applications on service peers are written in C++ to have better performances in signals analysis.

The presence recognition system is composed of two antennas: one inside and one outside the agency. An antenna reads the tags carried by the customers and sends a signal to the system. Each signal is filtered in order to reduce the redundancy of information and has associated to a timestamp. In this way the system is able to infer if a customer is entering or leaving a bank agency. For more technical information on the customers' presence recognition see [16].

5 Conclusions and Further Work

Adaptivity is gaining more and more the attention of the academic and industrial worlds. This affirmation is sustained by the increasing number of events (e.g., conferences, workshops) and publications (e.g., ACM TAAS journal, books, special issues)

having adaptivity as central topic, as well as by the various projects (e.g., Odyssey [9], ReMMoC (A Reflective Middleware to support Mobile Client Interoperability) [7], MobiPADS (Mobile Platform for Actively Deployable Service) [3], CARISMA (Context-Aware Reflective mIddleware System for Mobile Applications) [1]).

In this paper we have presented the main aspects of our architectural model for adaptive distributed systems focusing attention on the design of various types of strategies exploited to implement decisions at runtime. Three main types of strategies have been introduced. Strategies similar to the system services (service and QoS inspection, service discovery, service choice and service execution) have been defined also for domain services. Further work will be related to (1) the description of these strategies through a formal approach such as the one used in the context of the Rainbow project [5] and (2) the extension of the current set of defined strategies while considering other case studies.

We have described how our solution is used in the context of a case study. This case study is under development and we plan to extend it with further adaptive strategies and improve the already existent ones. For example, when a person enters the bank and he is not a customer then the staff members may be interested in convincing the person to become a customer. Or, when the director is notified that an important customer is present in the bank agency, we will consider in the adaptation process also the activity the director is currently performing in order to avoid disturbing him from another important task. We plan to address also customers with financial problems having important debts. The system identifies and proposes the best solution for this type of customers to overcome their financial problems. In this scenario besides the cashiers also one of the account consultants is notified.

Further work will be related to performance evaluations considering additional case studies. For example, we aim to consider also case studies similar to the personalized tourist guides inside a museum or in a city in order to address a wider range of issues related to the design of strategies for adaptivity.

References

1. Capra, L., Emmerich, W., Mascolo, C.: CARISMA: Context-Aware Reflective mIddleware System for Mobile Applications. *IEEE Transactions on Software Engineering* 29(10), 929–945 (2003)
2. Ceriani, S., Raibulet, C., Ubezio, L.: A Java Mobile-Enabled Environment to Access Adaptive Services. In: *Proceedings of the 5th Principles and Practice of Programming in Java Conference*, pp. 249–254. ACM Press, Lisbon (2007)
3. Chan, A.T.S., Chuang, S.N.: MobiPADS: A Reflective Middleware for Context-Aware Mobile Computing. *IEEE Transactions on Software Engineering* 29(12), 1072–1085 (2003)
4. Finkenzeller, K.: *The RFID Handbook – Fundamentals and Applications in Contactless Smart Cards and Identification*. Wiley & Sons LTD, Swadlincote (2003)
5. Garlan, D., Cheng, S.W., Huang, A.-C., Schmerl, B., Steenkiste, P.: Rainbow: Architecture-based Self-Adaptation with Reusable Infrastructure. *IEEE Computer* 37(10), 46–54 (2004)
6. Gorton, I., Liu, Y., Trivedi, N.: An extensible and lightweight architecture for adaptive server applications. *Software – Practice and Experience Journal* (2007)

7. Grace, P., Blair, G.S., Samuel, S.: ReMMoC: A reflective Middleware to Support Mobile Client Interoperability. In: Meersman, R., Tari, Z., Schmidt, D.C. (eds.) CoopIS 2003, DOA 2003, and ODBASE 2003. LNCS, vol. 2888, pp. 1170–1187. Springer, Heidelberg (2003)
8. McKinley, P.K., Sadjadi, S.M., Kasten, E.P., Cheng, B.H.C.: Composing Adaptive Software. *Computer* 37(7), 56–64 (2004)
9. Noble, B.: System Support for Mobile, Adaptive Applications. *IEEE Personal Communications*, 44–49 (2000)
10. Raibulet, C., Arcelli, F., Mussino, S., Riva, M., Tisato, F., Ubezio, L.: Components in an Adaptive and QoS-based Architecture. In: Proceedings of the ICSE 2006 Workshop on Software Engineering for Adaptive and Self-Managing Systems, pp. 65–71. IEEE Press, Los Alamitos (2006)
11. Raibulet, C., Ubezio, L., Mussino, S.: An Adaptive Resource Management Approach for a Healthcare System. In: Proceedings of the 19th International Conference on Software Engineering & Knowledge Engineering, Boston, Massachusetts, USA, pp. 286–291 (2007)
12. Raibulet, C., Arcelli, F., Mussino, S.: Exploiting Reflection to Design and Manage Services for an Adaptive Resource Management System. In: Proceedings of the IEEE International Conference on Service Systems and Service Management, pp. 1363–1368. IEEE Press, Los Alamitos (2006)
13. Raibulet, C., Arcelli, F., Mussino, S.: Mapping the QoS of Services on the QoS of the Systems' Resources in an Adaptive Resource Management System. In: Proceedings of the 2006 IEEE International Conference on Services Computing, pp. 529–530. IEEE Computer Society Press, Los Alamitos (2006)
14. Raibulet, C., Ubezio, L., Gobbo, W.: Leveraging on Strategies to Achieve Adaptivity in a Distributed Architecture. In: Proceedings of the 7th Workshop on Adaptive and Reflective Middleware (2008)
15. Song, J., Kim, H.: The RFID Middleware System Supporting Context-Aware Access Control Service. In: Proceedings of the 8th International Conference on Advances Communication Technology, vol. 1, pp. 863–867. IEEE Press, Los Alamitos (2006)
16. Ubezio, L., Valle, E., Raibulet, C.: Management of Multi-Services Structures through an Access Control Framework. In: Kaschek, R., et al. (eds.) UNISCON 2008. LNBIP 5, pp. 519–530. Springer, Heidelberg (2008)

Genetic Algorithm Application for Traffic Light Control

Ayad. M. Turkey¹, M.S. Ahmad¹, M.Z.M. Yusoff¹, and N.R. Sabar²

¹ Universiti Tenaga Nasional, Km 7, Jalan Kajang-Puchong,
43009 Kajang, Selangor, Malaysia

ayad_b2006@yahoo.com, {sharif,zaliman}@uniten.edu.my

² University Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia

Abstract. In this paper, we describe the design of an intelligent traffic light control based on genetic algorithm. This paper is part of our work in which we attempt to use genetic algorithm in traffic light control and pedestrian crossing. In our approach, we use four sensors; each sensor calculates the vehicle density for each lane. We developed an algorithm to simulate the situation of an isolated intersection (four lanes) based on this technology. We then compare the performance between the genetic algorithm controller and a conventional fixed time controller.

1 Introduction

The monitoring and control of vehicular traffic pose a major challenge in many countries. The escalating number of vehicles in the cities not only has a huge environmental impact, but also results in loss of lives on the road. This situation demands a comprehensive approach involving a system which coordinates the traffic controls for smooth flowing vehicles.

In this paper, we address the problem of increased vehicular flow on the road. There is an urgent need to optimize traffic control algorithms to accommodate the increase in vehicles in urban traffic which experience long travel times due to inefficient traffic light controls. Optimal control of traffic lights using sophisticated sensors and intelligent algorithms, especially in four way traffic junction, could be a possible solution.

In a conventional traffic light controller, the traffic lights change at constant cycle times which are clearly not optimal. The preset cycle time regardless of the dynamic traffic load only adds to the problem. We apply the genetic algorithm theory in the traffic control system to provide an intelligent green interval response based on dynamic traffic load inputs, thereby overcoming the inefficiency of the conventional traffic controllers. Such approach resolves the challenges when sensors placed at every lane in a four-way junction control read the density of vehicles to provide inputs for the algorithms.

2 Related Work

Pappis and Mamdani made the first known attempt to use fuzzy logic in traffic light control for a theoretical simulation study of a fuzzy logic controller in an isolated

signalized intersection (2+2 lanes, one-way intersection) [1]. They compare their fuzzy method to a delay-minimizing adaptive signal control with optimal cycle time. The fuzzy controller is equal to, or slightly better than, the adaptive method used for comparison.

Other attempts to use fuzzy logic in traffic light controls are made by Tan, Khalid and Yusof [2]; Niittymäki and Kikuchi [3]; Chen, May and Auslander [4]; Choi [5]; and Conde and Pérez [6].

3 Model Design for Traffic Light

We use our genetic algorithm controlled traffic light system on a four-junction two-way lane. We use four sensors; each sensor detects the vehicle density for each lane. The system calculates the green and red light times to be given for vehicles. It also calculates the vehicles queue behind the red light plus the time taken for each vehicle to arrive at its target destination in static and dynamic modes, i.e., if vehicle ID 4 comes from lane A goes to a destination in lane D1, the system calculates the time that it takes to travel from A to D1.

3.1 Variables

The variables for our system include the input, cellular automata, genetic algorithms operating parameters and the output.

The input variables are as follows:

1. Vehicle Density (VD): a measure of vehicles that pass through a green light.
2. Vehicle Queue (VQ): a measure of vehicles density created behind a red light.

The output variables are as follows:

1. Queue of the Vehicles (QV): Number of Vehicles behind the red light per second in static and dynamic modes.
2. The Duration (D) of travel for each vehicle to arrive at the target destination in static and dynamic modes.

3.2 Cellular Automata

One way of designing and simulating (simple) driving rules of cars is by using cellular automata (CA). CA use discrete partially connected cells that can be in a specific state. For example, a road-cell can contain a car or is empty. Local transition rules determine the dynamics of the system and even simple rules can lead to chaotic dynamics [7].

We use cellular automata algorithm in this paper because it allows us to represent significant events that occur during congestions such as traffic standstill, resume motion, return to standstill again, and so on.

3.3 The Model's Algorithms

In our algorithms, we establish the algorithm steps: initialize population, evaluate population, chromosomes selection and chromosomes recombination.

1. Initialize population: Each chromosome contains two genes, the first gene is red time and the other one is green time. We set the chromosomes population to 100. Chromosomes need to be encoded and this is connected to the problem that genetic algorithm is meant to resolve. We use binary encoding to encode the chromosomes. In binary encoding every chromosome is a string of bits 0 or 1 and it gives many possible chromosomes, even with a small number of alleles. See Figure 1 for an example of chromosomes with binary encoding.

Chromosome A	101100101100101011100101
Chromosome B	111111100000110000011111

Fig. 1.

2. Evaluate population: This provides a way to rate how each chromosome (candidate solution) solve the problem at hand. It involves decoding the chromosomes into the variable space of the problem and then checking the result of the problem using these parameters. The fitness is then computed from the result.

Crossover Fraction: With the crossover fraction=0.8, we used two point crossover operation performed on the parent's generation, the result of which is stored in a mean array. In this array, the parent's generation is merged with the children. These steps are repeated until the total number of the crossover operation is half the size of the initialization. We can then say that the crossover operation is completed.

Mutation Fraction: With the mutation fraction=0.2, we performed this operation on the parent's generation. From the results in the mean array, a random number is generated and the result of comparison between this number and mutation fraction are determined by the occurrence or non-occurrence of mutations. These steps are repeated until the total number of mutation operations is half the size of the initialization. We can then say that the mutation operation is completed.

3. Chromosome selection: The chromosomes are selected for propagation to future populations based upon their fitness. Chromosomes which have high fitness value have a good chance to be chosen for future population. For selection of chromosomes, we use the "Roulette-wheel with probability of selection that is proportional to fitness" based upon the fitness of the chromosomes. See Figure 2.

We determine the fitness function to identify the solutions. The fitness function is calculated based on many parameters (queue, density, green light time and red light time). Our fitness function consists of two parts:

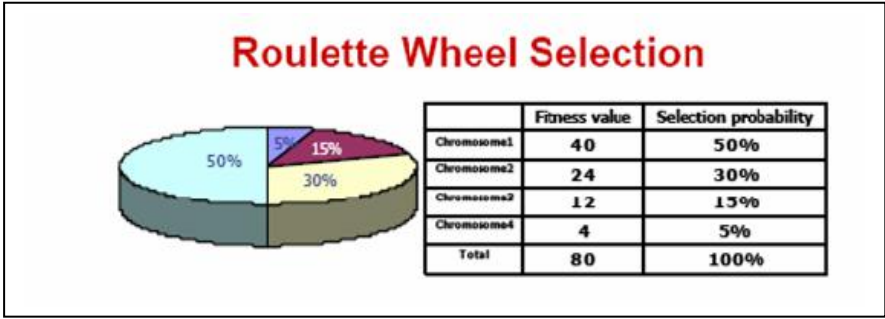


Fig. 2.

- a. We calculate the green time necessary due to the queue formed by the red light, $(VQ \times \text{vehicle\_time\_for\_passing})$ and compare this value with the past green times to obtain a good value for the green time. We set the $\text{vehicle\_time\_for\_passing}$ to 3 s.
- b. In the same way, we calculate the length of queue, which forms during the red time, $(VD \times \text{red})$, and the density of vehicles in the same lane at the same time. The quality of performance increases whenever this value decreases. We give greater attention to optimize the green time at the expense of queue length. Therefore, we multiply a ruling parameter, which change the fitness function as follows:

$$(VD \times \text{red})^3 - (\text{green} - VQ \times \text{vehicles\_time\_for\_passing})^2$$

4. Chromosome recombination: In recombination, pairs of chromosomes are recombined, possibly modified, and then placed back into the population as the next generation. The process continues again at evaluation until the problem represented by the chromosomes is solved, or some other exit criterion is met such as convergence, or the maximum number of generations is reached.

The next step in the operation is evaluating the generation to determine the resulting quality of these individuals compared with the previous generation. This is done by arranging the elements of the array (mean array) in increasing values provided by the fitness function.

Ordering the array elements in this way contributes to better identification of individual generations (parent and child generations). The first set of the elements of this array (mean array) is copied to the parent's array. These elements form 70% of the members of the new generation of parents. The rest (30%) is generated by using the random function. The algorithms read the new inputs after five generations to get good solutions.

4 Comparisons between Static and Dynamic Control

We compare the performance between the static control model (fixed cycle time) and our dynamic (genetic algorithm) control model. We use the same inputs for both models. In the static control model, we set the green and the red time for the vehicles

to 20 second for each lane. In this control model, the vehicles in one lane will wait even if there is no vehicles queue in the other lane.

In the dynamic control model, the times determined by genetic algorithm depend on the vehicles density (VD) and vehicles queue (VQ) parameters that are read from sensors for each lane resulting in synchronized green and red time. If there are no vehicles queue in one lane, the green time will be zero. Tables 1 and 2 below show the results of our experiments for both modes.

Table 1. Static and Dynamic Queue (Left) and Time Gain (Right)

Time (s)	No. of Vehicles (Behind a Red Light)		V-ID	Duration (s)		Time Gain (s)
	Static	Dynamic		Static	Dynamic	
1	0	0	1	6	5	1
2	0	0	2	16	5	11
3	0	0	3	6	6	0
4	3	2	4	16	7	9
5	4	1	5	16	4	12
6	6	1	6	16	6	10
7	6	1	7	7	8	-1
8	7	2	8	14	10	4
9	8	3	9	12	9	3
10	3	1	10	7	6	1
1	0	0	11	7	6	1

Table 2. Traveling time for Static Mode (Left) and Dynamic Mode (Right)

V-ID	Start Time	Arrival time	Duration (s)	V-ID	Start Time	Arrival time	Duration (s)
1	02:37:12	02:37:18	6	1	02:30:08	02:30:13	5
3	02:37:12	02:37:18	6	3	02:30:08	02:30:14	6
7	02:37:14	02:37:21	7	2	02:30:10	02:30:15	5
10	02:37:18	02:37:25	7	4	02:30:08	02:30:15	7
11	02:37:19	02:37:26	7	5	02:30:12	02:30:16	4
2	02:37:12	02:37:28	16	6	02:30:10	02:30:16	6
4	02:37:12	02:37:28	16	7	02:30:10	02:30:18	8
5	02:37:13	02:37:29	16	10	02:30:14	02:30:20	6
6	02:37:14	02:37:30	16	11	02:30:15	02:30:21	6
9	02:37:18	02:37:30	12	9	02:30:13	02:30:22	9
8	02:37:17	02:37:31	14	8	02:30:13	02:30:23	10

5 Conclusions and Future Work

From the results, we can say that the dynamic control model performs better than the static control model. Due to its flexibility, the dynamic control model is able to calculate the optimal green time based on vehicle density and queue length. Results also show

that significant time gain is experienced in traveling through the GA-controlled traffic light system.

In our future work we will extend the application of genetic algorithm in traffic light control systems in which a pedestrian crossing is included. We will address all the likelihood that could happen at a four-way traffic junction for vehicles passing and one lane for pedestrian crossing.

References

- [1] Pappis, C.P., Mamdani, E.H.: A Fuzzy Logic Controller for a Traffic Junction. *IEEE Transactions Systems, Man, and Cybernetics* SMC-7(10), 707–717 (1977)
- [2] Tan, K.K., Khalid, M., Yusof, R.: Intelligent Traffic Lights Control by Fuzzy Logic. *Malaysian Journal of Computer Science* 9(2), 29–35 (1996)
- [3] Niittymäki, J., Kikuchi, S.: Application of Fuzzy Logic to the Control of a Pedestrian Crossing Signal. *Transportation Research Record: Journal of the Transportation Research Board* 1651, 30–38 (1998)
- [4] Chen, L.L., May, A.D., Auslander, D.M.: Freeway Ramp Control Using Fuzzy Set Theory for Inexact Reasoning. *Transportation Research, Part A* 24(1), 15–25 (1990)
- [5] Choi, W., Yoon, H., Kim, K., Chung, I., Lee, S.: A traffic light controlling FLC considering. In: Pal, N., Sugeno, M. (eds.) *AFSS 2002. LNCS*, vol. 2275, pp. 69–75. Springer, Heidelberg (2002)
- [6] Conde, C., Pérez, J., González, P., Silva, J., Cabello, E., Monclús, J., Santa Cecilia, T.: A Conflict-Avoiding, Artificial Vision Based, Intelligent Traffic Light Controller
- [7] Nagel, K., Schreckenberg, M.: A cellular automaton model for freeway traffic. *J. Phys.* (1-2), 2221–2229 (1992)

Weaving Business Processes and Rules: A Petri Net Approach

Jian Yu¹, Quan Z. Sheng¹, Paolo Falcarin², and Maurizio Morisio²

¹ School of Computer Science, The University of Adelaide,
Adelaide, SA 5005, Australia

jian.yu01@adelaide.edu.au, qsheng@cs.adelaide.edu.au

² Dipartimento Automatica e Informatica, Politecnico di Torino,
Corso Duca degli Abruzzi 24, 10129 Torino
{paolo.falcarin,maurizio.morisio}@polito.it

Abstract. The emerging service-oriented computing paradigm advocates building distributed information systems by chaining reusable services instead of by programming from scratch. To do so, not only business processes, but also business rules, policies and constraints need to be encoded in a process language such as Web Services Business Process Execution Language (WS-BPEL). Unfortunately, the intermixing of business processes and rules in a single process weakens the modularity and adaptability of the systems. In this paper, we propose a formal approach to model the weaving of business processes and rules, following the aspect-oriented principle. In particular, we use Predicate/Transition (PrT) nets to model business processes and business rules, and then weave them into a coherent PrT net. The resulting woven nets are ready for analysing system properties and simulating system behaviour.

Keywords: Business process modelling, business rules, aspect-orientation, Petri nets.

1 Introduction

Service-oriented computing (SOC) builds on the software engineering trends of greater encapsulation and composing rather than programming [1]. It's why business processes play a central role in SOC: using distributed, platform-independent, and well-encapsulated services as basic building blocks, business processes organize and coordinate the behaviour of services to achieve business goals in a loosely coupled and flexible manner. In the case of Web services, Web Services Business Process Execution Language (WS-BPEL, BPEL for short) has been considered as a de facto industry standard to create composite service applications.

However, the flow logic encoded in processes cannot represent the complete features of a business: there are also rules, policies and constraints manifesting the decision aspect of the business. In fact, business rules are used extensively in some decision-intensive domains such as finance and insurance sectors to model

and document business requirements. A serious problem appears if we implement business rules using a process-oriented paradigm where business rules are mixed and coded in the process logic as a whole monolithic block. As a result, the original modularity of business rules is lost and it becomes hard for business rules to change without affecting the core composition logic [2].

In this paper, we use Predicate/Transition nets (PrT nets) [3], a kind of widely used high-level Petri nets, to model both business processes and business rules, and then use an aspect-oriented mechanism to weave them into a coherent PrT net. Our approach not only keeps the modularity of business rules, but also supports formal verification thanks to the well-established theoretical foundation of Petri nets.

The rest of this paper is organized as follows: Section 2 briefly overviews some fundamental concepts and definitions that underpin the discussion throughout the paper. Section 3 explains how to model business rules with PrT nets. Section 4 explains the PrT net-based aspect and the weaving mechanism, and Section 5 concludes the paper.

2 Business Rules and Aspect-Orientation

In this section, we briefly review the concepts of business rule and aspect-orientation. Some example rules and aspects used throughout the paper are also discussed.

According to the Business Rules Group [4], a business rule is a statement that defines or constrains some aspect of a business. It is intended to assert business structure or to control the behaviour of the business. In [5], business rules are classified into four types: *constraint rule*, *action-enabler rule*, *computation rule*, and *inference rule*.

For instance, a travel-package-requesting scenario could have the following business rules [2]:

R_1 (*constraint rule*): a vacation request must have a departure airport and a destination airport.

R_2 (*action-enabler rule*): if no flight is found, do not look for accommodation.

R_3 (*computation rule*): if more than 2 persons travel together, give 10% discount to the total price.

R_4 (*inference rule*): if a customer is frequent customer, he gets a discount of 5%.

The reason why R_4 is an inference rule is that to resolve what is a frequent customer, we need another two rules:

R_5 (*constraint rule*): if a customer has bought more than 5 travel packages, he is a frequent customer.

R_6 (*constraint rule*): if a customer has bought products for a sum exceeding 4000 euros, he is a frequent customer.

Aspect-orientation is a strand of software development paradigm that models scattered crosscutting system concerns as first-class elements, and then weaves them together into a coherent model or program.

Referring to Aspect4J [6], an aspect-oriented program usually consists of base modules and a number of aspects that modularizes crosscutting concerns. An *aspect* wraps up *pointcuts* and *advices*. A pointcut picks out certain *join points*, which are well-defined points (e.g., method calls) in the program flow. An advice is a piece of code that is executed when a join point is reached. There are three types of advices:

- A *before advice* runs just before the join points picked out by the pointcut.
- An *after advice* runs just after each join point picked out by the pointcut.
- An *around advice* runs instead of the picked join point.

Following gives an example to illustrate the idea of aspect-orientation. Supposing we have a simple business process containing three sequentially executing services, say *getFlight*, *getAccommodation*, and *calculatePrice*, to apply all the business rules ranging from R_1 to R_4 on this process, we first define the three services as pointcuts so that whenever a service is called, a join point is reached where advices can take effect. Then we define four advices: R_1 *before* *getFlight*, R_2 *around* *getAccommodation*, R_3 *after* *calculatePrice*, and R_4 *after* *calculatePrice*. The above pointcuts and advices can be grouped as an aspect. This aspect means: R_1 should be satisfied before executing *getFlight*; and if the condition of R_2 is true, *getAccommodation* will not be executed; and R_3 and R_4 should be executed after *calculatePrice*.

3 Modeling Business Rules with PrT Nets

In this section, we show our idea of how to use PrT nets to model the above-mentioned four types of business rules. Specifically, we use the terms *constraint nets*, *action-enabler nets*, *computation nets* and *inference nets* to call the nets representing corresponding business rules, and call them *rule nets* in general.

Syntactically, we classify the transitions in a rule net with three stereotypes:

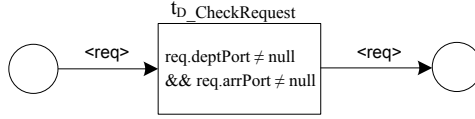
- An *action transition* T_A represents a business activity, e.g., get flight.
- A *computation transition* T_C represents the specific action of assigning variables with arithmetic expressions, e.g., assigning $price \times 90\%$ to the variable *price*.
- And a *dummy transition* T_D does nothing.

Transition stereotypes provide additional information to a net at the business level; they do not change the behavioural semantics of rule nets.

To model a constraint rule, we use a dummy transition with its inscription representing the constraints. For example, R_1 can be modelled as the constraint net in Fig. 11.

To model an action-enabler rule, we use an action transition to represent the business activity, and the inscription of this transition representing the enabling condition.

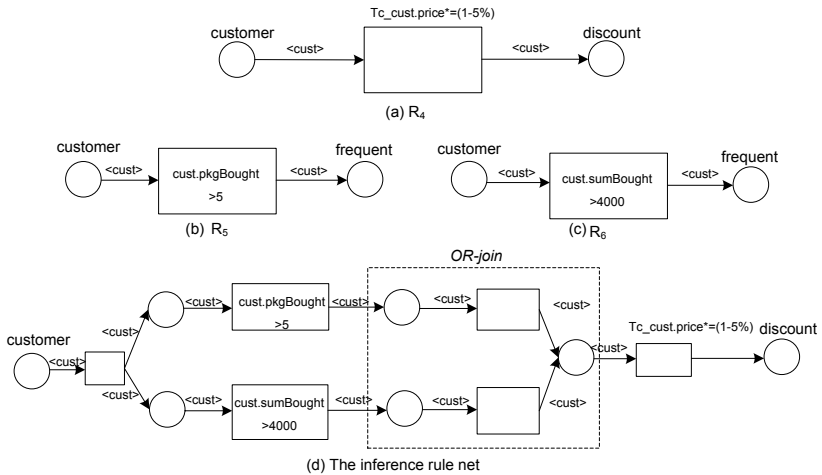
To model a computation rule, we use a computation transition with its name representing the computation expression. Note that the computation expression

Fig. 1. Constraint net R_1

is only reflected in the name of the transition since just like the execution of a business action, the interpretation of expressions is outside the scope of PrT nets.

An inference rule is created by composing two rule nets with AND-JOIN, OR-JOIN, or SEQ-JOIN operators. Intuitively, if we connect two rule nets with AND-JOIN, then the composed rule net means the conjunction of the two rules; if we connect two rule nets with OR-JOIN, then the composed rule net means the disjunction of the two rules. Note that AND-JOIN and OR-JOIN are also two workflow patterns defined in [7]. The sequential join of two rule nets means that the resolve of one rule net depends on the consequents of the other rule net. The *SEQ-JOIN* operation fuses the source place of the dependent net with the sink place of the independent rule net. Usually, an inference net can be built first by introducing the net representing the final goal, and then introducing the rules backwards with the *SEQ-JOIN* operation based on the cause-effect relations until all the newly introduced rules are resolvable.

Taking R_4 as an example, to resolve the meaning of frequent, we need another two rules: R_5 and R_6 . Fig. 2a, b and c are the rule nets for R_4 , R_5 and R_6 respectively. Because R_4 depends on the consequent of R_5 or R_6 , we use OR-JOIN to compose them and then use *SEQ-JOIN* to connect the combined consequent to the rule net of R_4 to form a complete inference rule net as depicted in Fig. 2d.

Fig. 2. The inference rule derived from R_4 , R_5 , and R_6

4 Weaving Rule Nets into Process Nets

Corresponding to rule nets, we use the term *process nets* to call the PrT nets representing business processes. For example, Fig. 3 is the process net N for the travel package request process described in Section 2.

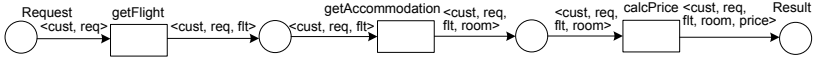


Fig. 3. Travel package request process net N

Just like aspect-oriented programming, we use *pointcut* to select a transition as join point. Without losing generality, every pointcut can only select one transition to simplify the definition of weaving. The advices are represented by rule nets. A rule net can be woven into a process net either *before*, *after*, or *around* a pointcut/transition. Another important concept is context net, for exposing context of the process net to the rule net. It is interesting to note that the process net and its rule nets could be authored independently, which reflects the modular feature of aspect-orientation. To weave them smoothly, we use context net

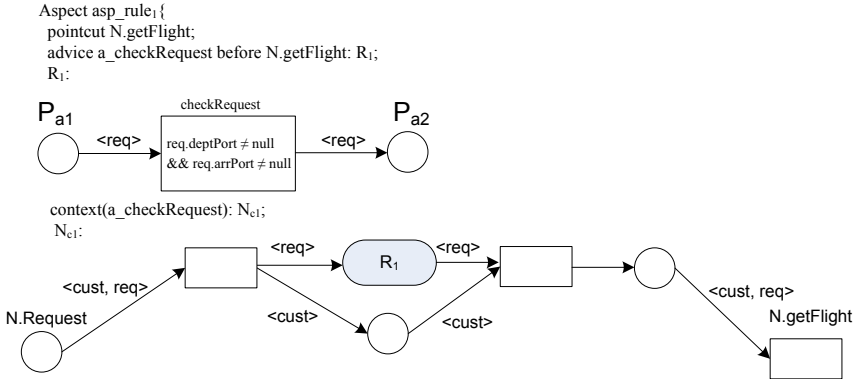


Fig. 4. Aspect asp_rule_1

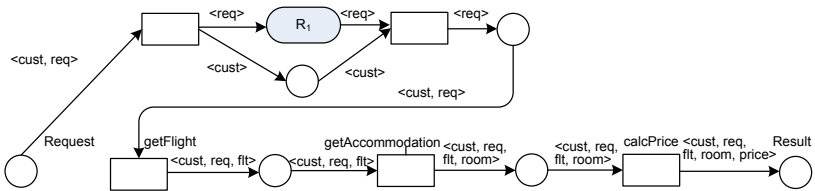


Fig. 5. Resulting net of weaving N and asp_rule_1

to put them in the same context and mediate possible parameter inconsistencies. For example, the input parameters for the transition $N.getFlight$ is a structure $\langle Customer, Request \rangle$, but its constraint net R_1 accepts $\langle Request \rangle$ as input. Finally, we can wrap up pointcuts, advices, and context net into an *aspect*.

To demonstrate our idea, we give an example on how to weave the constraint net R_1 before $N.getFlight$. Fig. 4 presents an aspect asp_rule_1 . The context net is built first by introducing the pre-set of the pointcut, i.e. transition $N.getFlight$, as the initial context, and then splitting the request to match the input of the advice net, finally merging the results and transferring them back to the cutting point. Note that we use an round rectangle to represent the net R_1 for simplicity.

Fig. 5 is the resulting net after we weave asp_rule_1 into N . This net is built simply by putting the context net and the process net together, and then deleting the arcs between the cutting point and its preset because it is a *before* cut. Note that elements with the same names are merged to reflect the expansion of context.

5 Conclusion

In this paper, we present a PrT net-based approach to weaving business processes and rules. The woven net not only keeps the modularity of rules, but also is ready for analysis and simulation by various PrT techniques and tools. We view our work presented in this paper as a first step towards a formalized model-driven approach of developing service-oriented information systems.

References

1. Yu, Q., Bouguettaya, A., Medjahed, B.: Deploying and Managing Web Services: Issues, Solutions, and Directions. The VLDB Journal 17(3), 537–572 (2008)
2. Charfi, A., Mezini, M.: Hybrid Web Service Composition: Business Processes Meet Business Rules. In: 1st International Conference on Service Oriented Computing ICSOC, pp. 30–38 (2004)
3. Genrich, H.J.: Predicate/Transition Nets. In: Brauer, W., Reisig, W., Rozenberg, G. (eds.) APN 1986. LNCS, vol. 255, pp. 207–247. Springer, Heidelberg (1987)
4. The Business Rules Group: Defining Business Rules, What Are They Really?, <http://www.businessrulesgroup.org>
5. von Halle, B.: Business Rules Applied: Building Better Systems Using the Business Rules Approach. Wiley, Chichester (2001)
6. Gradecki, J.D., Lesiecki, N.: Mastering AspectJ: Aspect-Oriented Programming in Java. Wiley, Chichester (2003)
7. van der Aalst, W., van Hee, K.: Workflow Management: Models, Methods, and Systems. MIT Press, Cambridge (2002)

Modeling Actions in Dynamic Engineering Design Processes

Vadim Ermolayev¹, Natalya Keberle¹,
Eyck Jentzsch², Richard Sohnius², and Wolf-Ekkehard Matzke²

¹ Department of IT, Zaporozhye National University, Zhukovskogo 66, 69063,
Zaporozhye, Ukraine

vadim@ermolayev.com, nkeberle@gmail.com

² Cadence Design Systems, GmbH, Mozartstr. 2 D-85622 Feldkirchen, Germany
{jentzsch, rsohnus, wolf}@cadence.com

Abstract. The paper presents the approach for modeling actions in the dynamic processes of engineering design in microelectronics and integrated circuits domain. It elaborates the formal framework for representing processes, the states of these processes and process environments, the actions being the constituents of the processes. Presented framework is implemented as the part of PSI suite of ontologies and is evaluated using three different methods: user evaluation, formal evaluation, and commonsense evaluation following PSI shaker modeling methodology. The suite of PSI ontologies is used for representing dynamic engineering design processes in Cadence Project Planning Expert System software prototype.

Keywords: PSI, action, task, activity, environment, design system, performance, framework, ontology.

1 Introduction

As many experts in microelectronic and integrated circuits design point out (e.g., [1]), one of the main industrial challenges is the gap between the capability of design technology and the productivity of design systems. For example, the capability of the design technology to accommodate digital gates on a chip is growing much faster than the capability of design teams using this technology and corresponding design environments to produce these gates in their designs. The consequence is that the effort required to be spent for designing a typical microelectronic device is growing substantially. Therefore, tools and methodologies for improving the performance of design systems are very highly demanded by industry.

PSI project¹ aims at developing models, methodologies, and software tools providing for rigorous engineering treatment of performance and performance management. PSI performance modeling and management approach focuses on performance as a

¹ Performance Simulation Initiative (PSI) is the R&D project of Cadence Design Systems GmbH.

pro-active action. A fine-grained dynamic model of an engineering design process, comprising a semantically rich action model, and a design system is therefore developed. PSI approach considers that performance is embodied in its environment and is controlled by the associated performance management process.

An engineering design process is a goal-directed process of transforming the representations of a design artifact in stateful nested environments. An environment comprises design artifact representations, resources, tools, and actors who perform actions to transform design artifacts using tools, and consume resources. Actions are admissible in particular environment states and may be atomic or compound, state-transitive or iterative, dependent or independent on other actions. The components of an environment may generate internal events or may be influenced by external events. Events may have causal dependencies. An engineering design process is a problem solving process which goals, partial goals, and environments may change dynamically. A decision taking procedure is associated with each state to allow environments to adjust the process taking these changes into account. Decisions are taken by actors modeled by software agents.

PSI software tools are developed [2] for assisting project managers to make robust planning, monitoring, and management of their design projects aiming at reaching best possible performance. Grounded decisions in planning are based on the knowledge base of project logs accomplished in the past. These logs provide vast and finely grained records of the performance of accomplished projects and may be used for simulating the behavior of the design system in response to different influences. At project execution phase PSI software may be used for predicting the behavior of the design system in the future based on the record of the partially accomplished dynamic engineering design process (DEDP), the knowledge about its environment(s), and performance simulations.

The focus of this paper is the framework for modeling actions. The rest of the paper is structured as follows. Section 2 analyses the related work in process modeling emphasizing the ways to model dynamic processes and pointing out the advancement of the presented modeling approach. Section 3 presents the action modeling framework of PSI. Section 4 reports how the framework has been implemented as the part of PSI suite of ontologies, evaluated, and used in PSI software prototype. Finally, concluding remarks are given and our plans for future work are outlined in Section 5.

2 Related Work

The framework presented in this paper is for modeling change and adequately accounting for dynamics in the processes of engineering design. Fundamentally, research in representing, reasoning, and capturing knowledge about change and dynamics produced the plethora of premium quality results which can't be even listed here due to space limit. Instead, we point to [3] as an excellent reference source. We also mention several related sources for analyzing our contribution.

McCarthy and Hayes [4] were the pioneers in introducing a logical formalism which became a mainstream for commonsense reasoning and reasoning about change in particular – the Situation Calculus (SC). Several authors have further developed

this approach resulting in several Event Calculi (EC) [5, 6]. Most of them use linear time instead of branching time characteristic to the SC. A topical representative of a branching time logic approach is [5]. Our approach is particularly close to DEC [6] because DEC uses discrete linear time representation. In difference to the mentioned EC our framework uses discrete linear time and time intervals with fuzzy beginnings and endings [7]. This enhancement makes our representation of events [8] and actions more flexible and expressive. For all other desired representational capabilities like causality, event triggering, context sensitivity, delays in effects, concurrency, release from the law of inertia [9] we rely on [6]. Some of these have already been accounted for: causality, triggering, delays. Elaboration of the rest is planned for the future work. The mainstream of formal business process modeling and engineering today is using PSL [10], PDL [11] or their extensions. Unfortunately, these formal process modeling frameworks do not fully allow breaking down the diversity of the processes encountered in real life. This diversity may be characterized for example by Sandewall's taxonomy [9] of the basic features of the processes. This classification embraces highly predictable, normal, manufacturing processes at one side and stochastic ("*surprising*"²), structurally ramified, time-bound processes characteristic for design domain, on the other side of the spectrum.

Presented modeling framework and the PSI suite of ontologies are the follow-up of our results published in [12]. The DEDP modeling framework in its part of process modeling bases its approach on [15-17]. The advancements of PSI approach are: (i) a rich typology of actions; (ii) environmentalistic approach to model processes, actions, their dependencies comprising concurrency; (iii) a state model refined using decision making mechanism and requirement sensitivity; (iv) an explicit difference between events and actions.

To the best of our knowledge, existing frameworks do not specify the difference between events and actions, except stating that actions are a kind of events: "the most important events are actions, and for a program to plan intelligently, it must be able to determine the effects of its own actions..." [cf. 18]. Such a view underestimates the role of events which occur without the involvement of an actor and the influence of those events on the environments of actions. Indeed, if we consider a person accidentally falling out from a window, this event can hardly be qualified as an action – the person had no purpose for or intention of falling out. The refinement proposed in PSI [19, 20] is that processes (compound actions) subsume to events, while atomic actions do that not. Atomic actions are a specific kind of an instrument for agents to proactively apply changes to their environment(s).

Our analysis of the variety of foundational ontologies [14] has revealed that the best matching ontological foundation for DEDP modeling is DOLCE [15] and the most appropriate referential commonsense theory is SUMO [16] extended by WordNet [17]. The semantics of our representation of actions, events, and environments is aligned with DOLCE through the PSI Upper-Level ontology [14]. The concepts of PSI Process ontology are mapped to SUMO through PSI Upper-Level ontology and WordNet using subsumptions.

² A process is considered "surprising" if it is allowed that a *surprising* or *exogenous* event may cause a change that is not anticipated by the process script [11].

3 Action Modeling Framework of PSI

A DEDP is the process of goal-directed (pro-active) transformation of a design artifact. A DEDP usually begins with collecting the initial available inputs (like the requirements, the high-level specification), continues in a sequence of stages normally defined by the design system, and ends up with the design artifact in a form which meets the goal of the design. These stages are the actions applied to a design artifact. Actions are distinct because they affect a design artifact differently by applying different changes (transformations) or causing no tangible change at all. Actions may be grouped in different combinations like sequences, branching structures, alternative or concurrent paths.

3.1 Preliminaries

A design artifact (DA) is a tangible product that is being designed in a DEDP. A DA may be a single indivisible design object or a hierarchical composition of design objects having the same or different types. A DA, and every design object in a DA, is incrementally elaborated as the emerging set of its representations. A DA representation (a representation further on in the paper) is the implementation of a DA in a particular form, format, or notation in which the DA is used for a distinctive purpose.

There exists only a partial order among representation types and respective representations. The semantics of this partial order is that a representation R_m which precedes another representation R_n is more abstract, while R_n is more elaborated. The distance between two representations R_m and R_n may also be of interest. Indeed, a question about how much is R_n more elaborated (or more abstract) than R_m is important because the answer characterizes the difficulty of the transformation of a DA between those representations. Difficulty is understood as the amount of abstraction crossed by a transformation. As transformations are applied by actions, difficulty (and distance) is somehow reflected by the effort to be spent for an action.

In any combination, actions lead processes to particular states. The simplest possible DEDP may be described by specifying its initial (triggering) influence, its initial state, its action, its target state, and the change in the design artifact caused by the specified action and reached in the target state. A more complex DEDP may comprise both atomic and compound actions. Hence, in a general case, a DEDP description should also contain the specification of its intermediate states. A DEDP state is a state S of DEDP environment which is characterized by the set of the pre-requisites for the associated actions. These pre-requisites are either the events [8] which, if perceived as happenings [8], trigger influences that change the course of action, the representations which are required for an action, or their combination. A DEDP state is the state of affairs in which a decision to perform one of the admissible actions (for example, to cease the process) is to be made:

$$S = \langle \mathbf{E}, \mathbf{R}, D_s, \mathbf{A} \rangle, \quad (1)$$

where: $\mathbf{E}$ is the set of associated events, $\mathbf{R}$ is the set of associated representations, D_s is the mechanism to make the decision to take a certain associated action, $\mathbf{A}$ is the set of admissible actions.

A representation R which is unconditionally available after a state S is reached is the characteristic representation of this state and belongs to the set $\mathfrak{R}$ of characteristic representations of this state.

Further on, to find out if a representation is really the thing we wanted to receive, the characteristics of the representation are verified against the requirements. For simplicity reasons only independent characteristics are taken into account. An independent characteristic c is the property of a representation which may be measured as recommended by the design system independently of the other characteristics. The set of independent characteristics of a representation is denoted as $\mathbf{C} = \{c_1, \dots, c_n\}$. A requirement $\rho: \mathbf{C} \rightarrow \{\text{true}, \text{false}\}$ is the Boolean function of the independent characteristics of a representation. Provided that requirements are defined for a representation, the degree of the success of an action elaborating the representation could be measured. If an action was successful enough the corresponding state may be considered as achieved. Otherwise a corrective action should be taken to improve the result. Hence the difficulty of an action is the function of the requirements to its target representation.

Requirements to the same representation may differ in different phases of a DEDP. For example, the characteristic of the density of the elements on the integrated circuit becomes really important at a place and route phase, though accounted for more liberally at earlier phases. Hence, a requirement is the attribute of a representation in a certain DEDP State. Requirements may be changed when a DEDP is already being executed. Such (events of) late changes to the requirements may result in unpredictable changes in the DEDP. If $T = [0, T]$ is the life time of a DEDP then a dynamic requirement $\rho(t): \mathbf{C} \rightarrow \{\text{true}, \text{false}\}, t \in [0, T]$ is a requirement which may be changed during the life time of a DEDP, otherwise it is a static requirement.

Accounting for the requirements implies the changes in the model of a state. A ρ -sensitive state is a DEDP state in which representations are constrained by a non-empty set of requirements $\mathbf{P} = \{\rho_1, \dots, \rho_n\}$, which may be static or dynamic:

$$S = \langle \mathbf{E}, \mathbf{R}, \mathbf{P}, D_s, \mathbf{A} \rangle. \quad (2)$$

One important kind of a requirement is a quality requirement. Such requirements are based on the characteristics measured as prescribed by the used quality model.

3.2 Action Kinds

While modeling actions it is important to pay attention to the following characteristic features: (i) is an action simple or compound? (ii) does an action transit a DEDP to a different state? (iii) what are the changes applied to the DA? (iv) what are the dependencies among certain actions?

Compound and Atomic Actions. Actors may have different understanding of the actions in a DEDP (Fig. 1). Some of them, according to their role, prefer to operate high-level actions. For example, a project manager may find more appropriate to specify only high-level actions in the project plan, like *front-end design* and *back-end design*. The others may tend to go deeper in the details of the actions. A front-end designer will definitely notice that a high-level *front-end design* action comprises

several lower-level actions like *RTL development*, *testbench development*, etc. Therefore one may deduce the hierarchical structure of the actions in a DEDP³ from the different facets of understanding a DEDP by different actors playing different roles. It is however important to find out if there are the basic building blocks for these actions – the ones which are understood as indivisible (atomic), by all the actors in a design system. It is rational to consider that such atomic actions exist and are defined by the design technology used within the design system. Such atomic actions are further on referred to as activities.

Definition 1: *an activity.* An activity is a basic indivisible (atomic) action which is allowed, supported, and provided by a design technology. An activity is the only action which is executed and applies the atomic chunk of the transformation to a design artifact.

Compound actions are the parts of a DEDP which are shrunk into one edge for convenience and the proper representation for different roles. These composite actions are denoted as tasks.

Definition 2: *a task.* A task is a compound action which may be represented as the composition of the other tasks and activities. Such representations are different in the knowledge of different actors.

As shown in Fig. 1, tasks may contain several transformation paths.

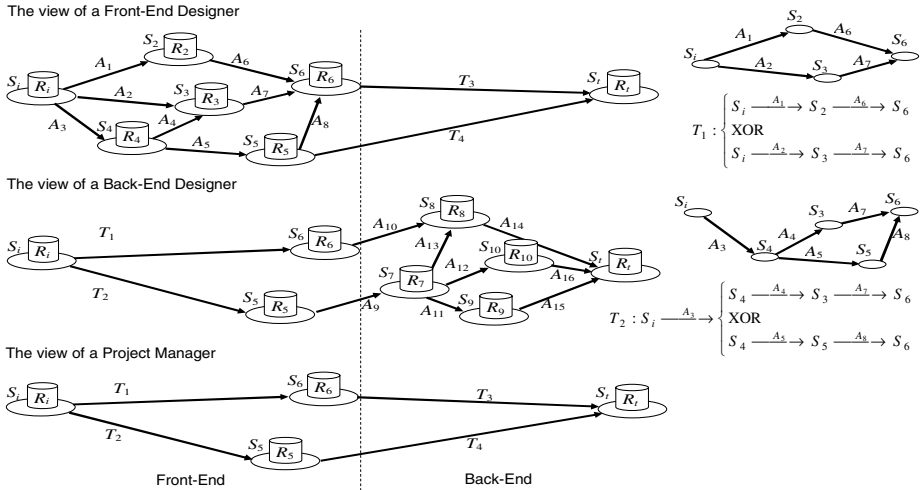


Fig. 1. Compound and atomic actions in subjective views of different actors

State-Transitive Actions. A DEDP transits to a new state when the pre-requisites for such a transition are met. These pre-requisites are: (i) events which indirectly trigger an action or (ii) the availability of characteristic representations for the target DEDP state. For clarity we shall consider the events occurring outside of DEDP environment

³ Like in a project plan. Indeed, the Actions in a project plan are often presented in a hierarchy.

and the events generated within the environment of a DEDP as separate kinds of events – external and internal ones.

An action will transit a DEDP to a new state S if representations produced by this action “complete” the set $\mathfrak{R}$ of the characteristic representations of S . According to the classification of the results of actions, the following types of actions may result in DEDP state transition because they produce new representations: productive actions, decomposition and integration actions.

It can not be expected that external events occur in a controllable manner. Therefore unexpected happenings and appropriate reactions to them in the form of influences should be accounted for. External events may or may not be perceived by the actors in the DEDP environment. External events may cause environmental changes of different magnitude. We shall say that an environment is stable with respect to a particular external event if the change caused by this event is negligibly subtle. On the contrary the environment is not stable with respect to a particular external event if the magnitude of the incurred change is substantial. By saying “substantial” we mean that the magnitude of the change⁴ requires that a corrective action is applied to the DEDP. In the latter case it is important to ensure that such an event is perceived and the influence is generated to execute required corrective actions. For the sake of uniformity and simplicity we shall consider a forced change of DEDP state as the only possible type of a corrective action.

Internal events are generated by the components of the environment [8] of a DEDP – the design system. One possible kind of an internal event is that an actor executing an action becomes unavailable. Possible reactions are: (i) action suspension until the actor becomes available; (ii) actor substitution – no influence to DEDP; (ii) corrective action changing DEDP state and choosing a different transformation path executed by a different actor. Another possible internal event could be that a resource being consumed in the action becomes unavailable. Possible reactions are: (i) action suspension until the resource becomes available; (ii) resource substitution – no influence to DEDP; (iii) corrective action changing DEDP state and choosing a different transformation path where the resource is not consumed. Yet one more kind of an internal event is that the requirements to a representation have no chance to be met if the chosen iterative action is continued. Possible reaction is rolling back to the previous DEDP state and choosing a different transformation path.

If a DEDP has reached its target state S_i and all the requirements to the characteristic representations of S_i are met, a cessation action terminating the DEDP in success has to be applied. In all other cases either a change in the environment (like actor substitution or resource substitution) is sufficient or an action is decided to be suspended. Otherwise, a corrective action should be taken to choose a different transformation path in the process.

Corrective Actions. In a DEDP some transformation paths may be more risky than the others. Indeed, when for example a design system transits to a new design technology, the correlations among the requirements are not very well understood. The assessments of the quality provided by actions are not well grounded. If these settings are complicated by the dynamic factors, an action on a chosen transformation path

⁴ Corresponding thresholds should be found out experimentally when calibrating the model of the design system.

may unexpectedly result in missing the requirements to a characteristic representation. Though iterative actions may help further elaborating or refining the representation, there might be a situation in which the refinement is not longer possible using available actions. A corrective action may improve such a situation by: (i) rolling-back the transformation path to the nearest successful state or (ii) choosing the next-most productive transformation path as the back-up plan. Corrective actions may also be used as a mechanism of collecting facts on bad experience to make risk assessments more grounded in the future.

Iterative Actions. Some types of actions will iterate a DEDP in a ρ -sensitive state S if characteristic representations of S do not meet the set of requirements $P = \{\rho_1, \dots, \rho_n\}$ of state S , i.e. at least one of the requirement functions ρ is *false*. Moreover, only such types of actions may be triggered by D_S until the requirements are met. These types of actions are: refinement actions, elaboration actions, debugging actions, verification actions. A D_S may detect that several iterative actions should be performed at a certain phase of iterations at state S . By the analogy to productions-based inference engines these actions may be considered to be in a conflict set. The conflicts may be resolved by: (i) analyzing the dependencies among the actions in the conflict set; and (ii) assigning priorities to the (types of) independent actions.

Cessation Actions. The difference between a corrective action and a cessation action is that a corrective action transits a DEDP to its state, though different from the previous one, but a cessation action terminates a DEDP – i.e., moves it out of the state space. A cessation action may terminate a DEDP in success or in failure.

3.3 Dependencies and Concurrency

As emphasized by many authors, for example [18], best performance is achieved when the best achievable degree of coherence among the actions within a process is granted. Coherence among actions means several important things: (i) coherence in individual goals of different actors performing different actions in one process; (ii) proper distribution of the consumption of available resources in different actions; (iii) balance in the capacities of the tools used in different actions; (iv) appropriateness of the skills of the actors to the requirements of the actions assigned to these actors; (v) proper scheduling of the actions which use the results produced by other actions. All these aspects represent dependencies among actions. Hence, achieving coherence among these actions can be reached by coordination, which is the routine of “managing the interdependencies between activities” (cf. [19]). Therefore, a model of action dependencies should account not only for the direct dependencies of actions on the results of other actions (the latter case (v)), but for a broader variety of indirect dependencies (at least cases (i)–(iv)) revealed through the process environment. Generally speaking, action A_3 depends on another action A_1 when the post-effects of A_1 change the pre-requisites of A_3 . This dependency may be denoted in the terms of a DEDP State, an event, a happening, and an influence.

Let S_1, S_2 be ρ -sensitive States (2).

Definition 3: an action-related part of DEDP environment. $\Sigma|_{A_1} = \{R_1, AC_1, RT_1\}$ is the part of DEDP environment related to A_1 if:

- The execution of any action A associated with S_1 generates internal events changing the constituents of Σ_1 , or
- The occurrence of any external event E changing $\Sigma|_{A_1}$, if perceived by a member of AC_1 , may change the course of actions associated with S_1 by influencing these actions at their execution time

The constituents of $\Sigma|_{A_1}$ are (Fig. 2):

$R_1 = \{R_1, R_2, \dots, R_4\}$ – the set of representations available in S_1 and (if any) produced to reach S_2

$AC_1 = (AC_1, AC_2, \dots, AC_m)$ – the set of actors capable of executing A_1 associated with S_1 and available at the time when A_1 has to be commenced

RT_1 – the pool of resources consumed by and tools used in A_1 associated with S_1

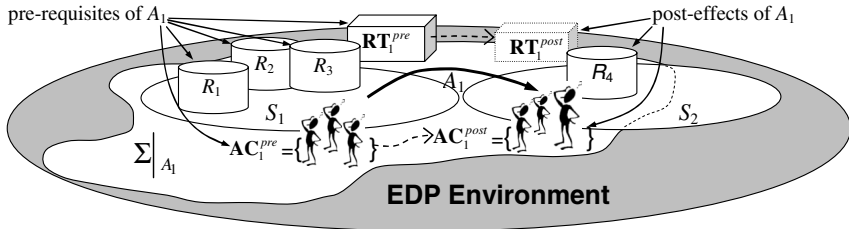


Fig. 2. A part of DEDP environment ($\Sigma|_{A_1}$) related to action A_1

It is important to note that, according to definition 3, if an action (A_1) causes state transition (S_1 to S_2), then $\Sigma|_{A_1} = \Sigma|_{A_1}^{pre} \cup \Sigma|_{A_1}^{post}$ comprises both the pre-requisites of A_1 and the post-effects of A_1 . Some of these pre-requisites are not associated with DEDP states, like AC_1^{pre} and RT_1^{pre} . Others belong to S_1 , like $R_1^{pre} = \{R_1, R_2, R_3\}$. Some of the post-effects are also state-independent: AC_1^{post} and RT_1^{post} containing those elements that have been changed by A_1 . These changes may be in the availability of actors, resources and tools and the capability of actors. Other post-effects belong to the target DEDP state S_2 : $R_1^{post} = \{R_4\}$. These post-effects also contain changed representations only.

If A_1 does not cause the transition of the process to a different DEDP state, the configuration of $\Sigma|_{A_1} = \Sigma|_{A_1}^{pre} \cup \Sigma|_{A_1}^{post}$ still remain similar to the previous case. The only difference is that $R_1^{post} = \{R_4\}$ belongs to the same DEDP state (S_1).

Let A_1 be an action causing the transition of the DEDP to DEDP state S_2 , A_3 be an action associated with DEDP state S_3 .

Definition 4: *a dependency.* A_3 depends on A_1 , if $\Sigma|_{A_3}^{pre} \cap \Sigma|_{A_1}^{post} \neq \emptyset$, otherwise A_3 and A_1 are independent.

Following [20] we shall classify dependencies as *weak* and *strong*. Action A_3 is strongly dependent on action A_1 if the execution of A_3 could not be started until A_1 is accomplished and its goal is fully met. Having in mind that the goal of an action in a DEDP is reaching the DEDP state in which all the required representations are made available, we may denote strong dependency as follows.

Definition 5: *a strong dependency.* A_3 strongly depends on A_1 if $\Sigma|_{A_3}^{pre} \cap \Sigma|_{A_1}^{post} = \Sigma' \subseteq \mathbf{R}_1^{post}$

Please note that definition 5 holds true also if A_1 is not a state transitive action.

Definition 6: *a weak dependency.* A_3 weakly depends on A_1 if:

- (i) $\Sigma|_{A_3}^{pre} \cap \Sigma|_{A_1}^{post} = \Sigma' \subseteq \Sigma|_{A_1}^{post} \setminus \mathbf{R}_1^{post}$ – *environmental dependency*, or
- (ii) $\Sigma|_{A_3}^{post} \cap \Sigma|_{A_1}^{post} = \Sigma' \subseteq \mathbf{R}_3^{post}$ – *facilitation dependency*

According to definition 6 environmental dependencies (i) are caused by sharing the pool of actors, the pool of tools, consuming the same resources, or by the combination of these reasons, normally indicating that the actions are competitive. On the contrary, facilitation dependencies (ii) indicate that actions are cooperative. Indeed, the interpretation of (ii) is as follows: A_1 facilitates A_3 in reaching its goal because it elaborates some part of the set of representations $\mathbf{R}_3^{post}$.

Accounting for action dependencies may help building better DEDP schedules thus improving their performance properties.

Concurrency among actions is one more aspect which may influence temporal properties of DEDP performance. It is evident that gaining more concurrency among the actions in a DEDP may result in shorter schedules and shorter execution times. Even partial overlaps in time intervals of action execution may optimize the overall performance. Unfortunately, it is not possible to execute all the actions in a DEDP in parallel or partly in parallel because of their dependencies.

Let, according to [7], I_{A_1} be the fuzzy time interval of the execution of A_1 and I_{A_3} be the fuzzy time interval of the execution of A_3 . Then the following definitions of *full* and *partial* concurrency among two different actions hold true.

Definition 7: *full concurrency.* Action A_1 is fully concurrent to action A_3 if $\text{Same}(I_{A_1}, I_{A_3}) \vee \text{Within}(I_{A_1}, I_{A_3})$.

Definition 8: *partial concurrency.* Action A_1 is partially concurrent to action A_3 if $\text{Overlaps}(I_{A_1}, I_{A_3})$.

In terms of action dependencies we may rightfully state that if action A_1 is independent to action A_3 then we may schedule and execute them fully concurrently. Granting concurrency to dependent actions is not that straightforward. The case of a strong dependency is simpler.

Corollary 1: *concurrency of strongly dependent actions.* If action A_3 is strongly dependent on action A_1 then they can not be executed concurrently.

The case of a weak dependency is more complex.

Corollary 2: *concurrency of environmentally dependent actions.* If action A_3 is environmentally dependent of action A_1 , then their concurrent execution, while being possible, may make the overall performance less optimal.

Corollary 3: *concurrency of actions with facilitation dependency.* If action A_1 facilitates the execution of action A_1 , then their concurrent execution while result in better overall performance.

4 Implementation and Evaluation

The action modeling framework presented above has been implemented as several OWL-DL⁵ modules of the PSI suite of ontologies. The initial implementation has been done in the suite of ontologies v.2.0. In this revision actions were modeled by the Task-Activity ontology and several relationships to the Actor ontology, Project ontology, Design Artifact ontology. In v.2.1 the ontological model of actions has been refined by introducing the associations to the Time ontology [7] and the Environment, Event, and Happening (E2H) ontology [8]. Finally, in v.2.2 the Upper-Level ontology [14] has been introduced and the modular structure has been refined by splitting the action model at the domain level into the Process Pattern ontology and the Process ontology. PSI shaker modeling methodology [14] has been used for refining the suite of ontologies and producing v.2.1 and v.2.2.

Fig. 3 pictures the UML diagram of PSI Process ontology v.2.2. Full details of this ontology are given in [21]. Uncolored packages in Fig. 3 represent different PSI core ontologies: Actor ontology; Project ontology; Design Artifact ontology; Environment, Event, and Happening ontology [8]; Time ontology [7]. The grey colored package represents the Resource extension ontology developed in the PRODUKTIV+ project⁶.

PSI Process ontology has been evaluated as the part of the core of PSI suite of ontologies v.2.2 as suggested by the shaker modeling methodology. Three different evaluation activities have been performed: (i) user evaluation; (ii) formal evaluation; (iii) commonsense evaluation.

User evaluation has been performed as a goal-based evaluation of the adequacy of the knowledge model and its implementation in the ontology to the set of requirements by the group of subject experts in microelectronic engineering design. It found out that the ontology adequately answers the competency questions formulated using the requirements by subject experts. Several test cases have been developed for evaluating the suite of ontologies using simulation. A testcase is a real or a fictive project for which at least all the initial data instances required for design system modeling are available. Acquiring a testcase allows to verify that the ontology is capable of storing all initial data as its instances and to start simulation using the multi-agent system

⁵ Web Ontology Language: <http://www.w3.org/TR/owl-guide/>.

⁶ PRODUKTIV+ is the R&D project funded by the German Bundesministerium für Bildung und Forschung (BMBF).

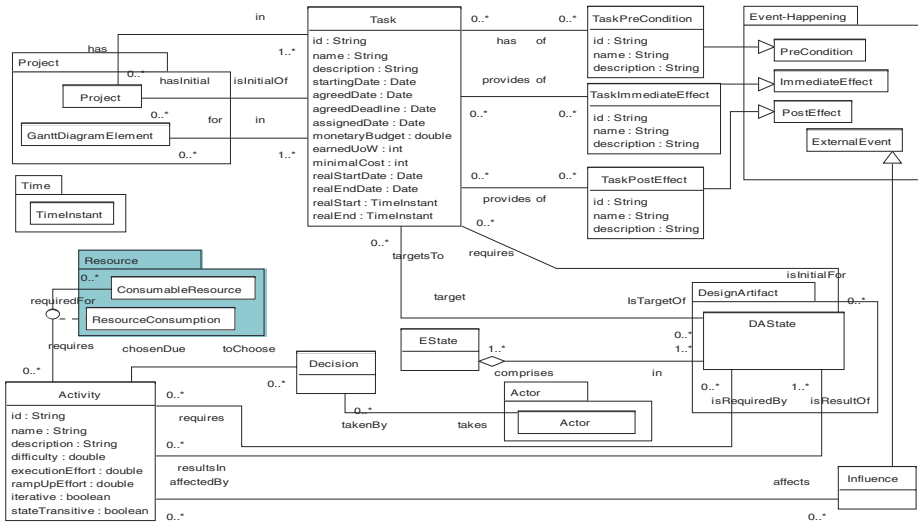


Fig. 3. UML diagram of the main concepts in the PSI Process ontology v.2.2

(MAS). Ideally, a testcase also provides a complete project execution record which is required for the calibration step and for the verification of the simulation results. Amongst the set of testcases are very simple and fictive ones describing tiny digital and tiny analog designs. These have been used to verify and refine the model and the ontologies. Others are based on real world projects of digital and analog chip design. These are fictive ones as well designed for demonstration purposes because real world project data usually may not be disclosed to public. For calibration the Project Planning Expert System MAS [2] was fed with the project definition and the knowledge base. Using these, it created a new work breakdown structure (Fig. 4). The result was then compared to the original structure, differences were analyzed, and corrections were made until the both roughly matched. At subsequent stages the MAS simulated project executions (Fig. 5) and again the results were compared to the original project course. Project log replay simulations and calibration experiments proved that the approach is effective and practical.

Formal evaluation has been performed using OntoClean methodology [22]. Its objective was to check the formal conformance of the taxonomy structure of the evaluated Suite of Ontologies v.2.2 to the meta-properties of rigidity, identity, and dependence [23]. Formal evaluation of PSI Process ontology together with PSI Upper-Level ontology [14] revealed that the taxonomy structure is conceptually correct.

The mappings of the concepts of the Process ontology to SUMO [16] through PSI Upper-Level ontology and WordNet [17] have been defined [13] for evaluating these ontologies with respect to the common sense [24]. This work revealed that the ontology adequately maps to SUMO – the chosen [14] commonsense reference ontology. This fact allows us believing that the process related part of the PSI suite of ontologies may be used not only internally in PSI and PRODUKTIV+ projects, but broader – as a descriptive theory of dynamically ramified processes in the domains which are

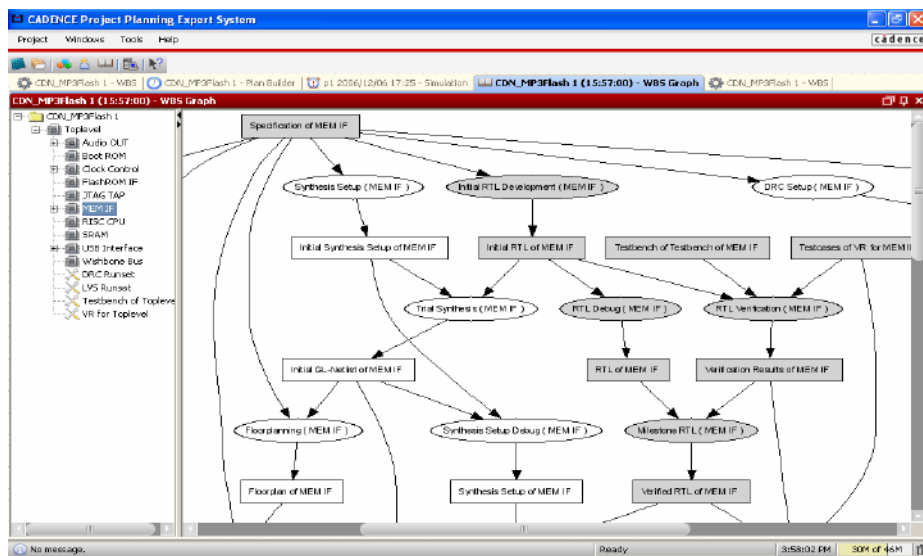


Fig. 4. The result of the planning phase of a design process simulation in Cadence Project Planning Expert System: generated Work Breakdown Structure graph. Ellipses stand for results, rectangles for tasks. Darker entries show the selected transformation path.

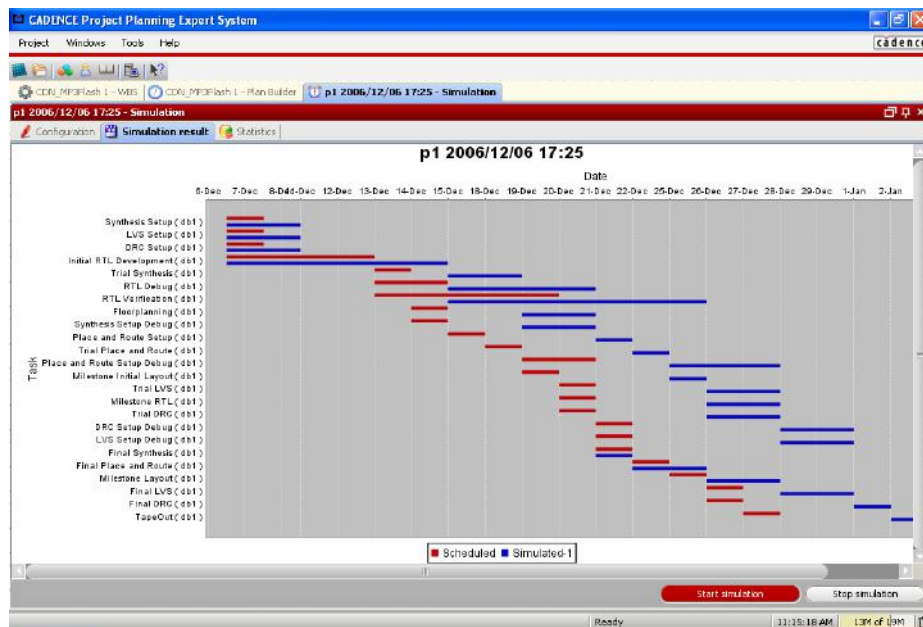


Fig. 5. The results of planning and execution phases of a design process simulation in Cadence Project Planning Expert System. Action durations are compared for planning (brighter bars) and execution (darker bars).

dynamic, complex, and non-deterministic similarly to engineering design. One good example is the domain of knowledge processes and knowledge workers investigated by the ACTIVE project⁷.

5 Concluding Remarks

The paper presents the action-related part of the PSI theoretical framework and its implementation as the Process ontology of the PSI suite of ontologies v.2.2. The advantages of the presented approach to modeling actions and processes are: (i) a rich variety of action kinds; (ii) environmentalistic approach to model processes, actions, their dependencies comprising concurrency; (iii) a state model refined using decision making mechanism and requirement sensitivity; (iv) an explicit difference between events and actions. These allow making process and action models being flexible and adaptive to the extent required for modeling structurally and dynamically ramified, time-bound processes characteristic for engineering design domain.

The presented ontological model of actions has been iteratively refined starting from its initial revision in v.2.0 of the PSI suite of ontologies till its current revision in v.2.2 using PSI shaker modeling methodology. The methodology subsumes several kinds of evaluation activities which have been performed as reported in the paper. Evaluation proved that the presented approach to modeling actions and processes is practical and effective. The core part and the several extensions of the PSI suite of ontologies are used in the Cadence Project Planning Expert System software prototype.

References

1. Van Staa, P., Sebeke, C.: Can Multi-Agents Wake Us from IC Design Productivity Nightmare? In: Mařík, V., Vyatkin, V., Colombo, A.W. (eds.) *HoloMAS 2007*. LNCS (LNAI), vol. 4659, pp. 15–16. Springer, Heidelberg (2007)
2. Sohnius, R., Jentzsch, E., Matzke, W.-E.: Holonic Simulation of a Design System for Performance Analysis. In: Mařík, V., Vyatkin, V., Colombo, A.W. (eds.) *HoloMAS 2007*. LNCS (LNAI), vol. 4659, pp. 447–454. Springer, Heidelberg (2007)
3. Mueller, E.T.: *Commonsense Reasoning*. Morgan Kaufmann Publishers, San Francisco (2006)
4. McCarthy, J., Hayes, P.J.: Some Philosophical Problems from the Standpoint of Artificial Intelligence. *Machine Intelligence* 4, 463–502 (1969)
5. Shafer, G., Gillett, P.R., Scherl, R.B.: The logic of events. *Ann. Math. Artif. Intelligence* 28(1–4), 315–389 (2000)
6. Mueller, E.T.: Event Calculus Reasoning through Satisfiability. *J. Logic and Computation* 14(5), 703–730 (2004)
7. Ermolayev, V., Keberle, N., Matzke, W.-E., Sohnius, R.: Fuzzy Time Intervals for Simulating Actions. In: Kaschek, R., Kop, C., Steinberger, C., Fliedl, G. (eds.) *UNISCON 2008*. LNBIP, vol. 5, pp. 429–444. Springer, Heidelberg (2008)

⁷ ACTIVE: Knowledge Powered Enterprise (<http://www.active-project.eu/>) is an Integrating Project funded by Framework Program 7 of the European Union.

8. Ermolayev, V., Keberle, N., Matzke, W.-E.: An Ontology of Environments, Events, and Happenings. In: Proc. 31st IEEE Annual International Computer Software and Applications Conference (COMPSAC 2008), pp. 539–546. IEEE Computer Society, Los Alamitos (2008)
9. Sandewall, E.: Features and Fluents. The Representation of Knowledge about Dynamical Systems, vol. 1. Oxford University Press, Oxford (1994)
10. Bock, C., Gruninger, M.: PSL: A semantic domain for flow models. *Software and Systems Modeling Journal* 4, 209–231 (2005)
11. Workflow Management Coalition. Workflow Standard. Process Definition Interface – XML Process Definition Language. V. 2.00, Doc. No. WfMC-TC-1025 (Final), October 3 (2005)
12. Ermolayev, V., Jentzsch, E., Karsayev, O., Keberle, N., Matzke, W.-E., Samoylov, V., Sohnius, R.: An Agent-Oriented Model of a Dynamic Engineering Design Process. In: Kolp, M., Bresciani, P., Henderson-Sellers, B., Winikoff, M. (eds.) AOIS 2005. LNCS, vol. 3529, pp. 168–183. Springer, Heidelberg (2006)
13. Ermolayev, V., Jentzsch, E., Keberle, N., Sohnius, R.: Performance Simulation Initiative. Meta-Ontology v.2.2. Reference Specification, tech. report PSI-ONTO-TR-2007-4, VCAD EMEA Cadence Design Systems GmbH (2008)
14. Ermolayev, V., Keberle, N., Matzke, W.-E.: An Upper-Level Ontological Model for Engineering Design Performance Domain. In: Li, Q., Spaccapietra, S., Yu, E., Olivé, A. (eds.) ER 2008. LNCS, vol. 5231, pp. 98–113. Springer, Heidelberg (2008)
15. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A.: WonderWeb Deliverable D18 Ontology Library (final). In: ICT Project 2001-33052 WonderWeb: Ontology Infrastructure for the Semantic Web (2003)
16. Niles, I., Pease, A.: Towards a Standard Upper Ontology. In: Guarino, N., Smith, B., Welty, C. (eds.) Int. Conf. on Formal Ontologies in Inf. Systems, vol. 2001, pp. 2–9. ACM Press, New York (2001)
17. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)
18. O'Donnell, F.J., Duffy, A.H.B.: Design Performance. Springer, London (2005)
19. Malone, T., Crowston, K.: Toward an interdisciplinary theory of coordination. Center for Coordination Science, Technical Report 120, MIT Sloan School of Management (1991)
20. Nagendra Prasad, M.V., Lesser, V.R.: Learning situation-specific coordination in cooperative multi-agent systems. *Autonomous Agents and Multi-Agent Systems* 2(2), 173–207 (1999)
21. Ermolayev, V., Jentzsch, E., Keberle, N., Sohnius, R.: Performance Simulation Initiative. The Suite of Ontologies v.2.2. Reference Specification. Technical report PSI-ONTO-TR-2007-5, VCAD EMEA Cadence Design Systems, GmbH (2007)
22. Guarino, N., Welty, C.: Supporting Ontological Analysis of Taxonomic Relationships. *Data and Knowledge Engineering* 39(1), 51–74 (2001)
23. Guarino, N., Welty, C.A.: A Formal Ontology of Properties. In: Dieng, R., Corby, O. (eds.) EKAW 2000. LNCS, vol. 1937, pp. 97–112. Springer, Heidelberg (2000)
24. Keberle, N., Ermolayev, V., Matzke, W.-E.: Evaluating PSI Ontologies by Mapping to the Common Sense. In: Mayr, H.C., Karagiannis, D. (eds.) Proc. 6th Int'l Conf. Information Systems Technology and its Applications (ISTA 2007). GI LNI, vol. 107, pp. 91–104. GI Bonn (2007)

Replenishment Policy with Deteriorating Raw Material Under a Supply Chain: Complexity and the Use of Ant Colony Optimization

Jui-Tsung Wong¹, Kuei-Hsien Chen², and Chwen-Tzeng Su²

¹ Department of International Business, Shih Chien University Kaohsiung Campus, Neimen Shiang, Kaohsiung, Taiwan, Republic of China
wongjt@mail.kh.usc.edu.tw

² Department of Industrial Engineering and Management, National Yunlin University of Science and Technology, Touliu, Yunlin, Taiwan, Republic of China
{g9321803, suct}@yuntech.edu.tw

Abstract. In order to enhance product competitiveness, customer satisfaction, and achieve quick response, it is common now for enterprises to become members of supply chains. In this paper, a replenishment policy for supply chain was optimized. It is a dynamic lot-sizing problem with deterioration. Most historical literatures consider deteriorations of end products. In practice, many raw materials possess significant deterioration effects, such as fishery and agricultural products. Therefore, this paper proposed a dynamic lot-sizing problem with deteriorating raw material. The point for this problem is to achieve a trade-off between excessive stocks and material deficiency. This paper proved that the proposed problem is NP-hard, and some properties of the proposed problem were also analyzed. Recently, techniques of artificial intelligence are becoming more and more mature, and are widely applied in various fields. In this paper, an ant colony optimization based on those properties was developed.

Keywords: Ant colony optimization, dynamic lot-sizing problem, deteriorating raw material, complexity.

1 Introduction

In order to face the arena of competition under the environment of global economics and trading, it is now a trend for enterprises to join the operations of supply chains. Generally, bullwhip effect exists in supply chains. Demand information is affected by forecast during supply chain transmission, and the demands are further distorted. Distortion effect is more prominent in upstream members. This is a key factor for long whip effect [1, 2]. Enterprises may lower bullwhip effect by information sharing [3]. This paper considered a replenishment policy that upstream member of a supply chain possesses power for controlling inventory levels. In this proposed problem, demand information is shared.

Inventory management is a key research field for supply chain. Lot-sizing problems are mainly categorized as continuous and discrete time-varying. In continuous time mode, economic order quantity method was proposed first. Rogers [4]

further considered multi-product economic lot-scheduling problem. Other continuous time literatures include: Moon et al. [5], and Jensen and Khouja [6]. In discrete time mode, since the solving of single-stage dynamic lot-sizing problem with dynamic programming by Wagner and Whitin [7], dynamic lot sizing problem is always a highly-discussed topic. In addition, researches for multi-stage lot-sizing problem include: Lee et al.[8], Jaruphongsa et al. [9], and Özdamar and Birbil [10]. However, the lot sizing of historical literatures above didn't consider the deterioration of products. Smith [11] proposed a single-stage dynamic lot-sizing problem, and products in his model will deteriorate after a constant period. Friedman and Hoch [12] proposed a single-stage dynamic lot-sizing problem with non-increasing deterioration rate. Hsu [13] proposed a single-stage dynamic lot-sizing problem with deteriorating products. His problem assumed that quantity of supply is unlimited, and the backordering is not allowed. He solved this problem by dynamic programming. Hsu [14] further discussed about the dynamic lot sizing problems for deteriorating products when backordering is allowed. Most of these related works assume that final products possess deterioration, and the supply quantity is unlimited.

In this paper, a dynamic lot-sizing problem with deteriorating raw material (DLSPDRM) was proposed. One of the features in the proposed problem is that deterioration rate before processing is different from the rate after processing. In practice, product after processing may become less likely to deteriorate (such as vacuum and harder packaging materials). In this paper, the complexity and some properties for this problem were analyzed. An ant colony optimization (ACO) was proposed to decide replenishment policies. The organization of this paper is as follows. In Section 2 the DLSPDRM was formulated. Section 3 discussed properties of the proposed problem, and constructed the ACO approach. In section 4 the performances of the two proposed ACO approaches were compared. In section 5 the result in this paper was discussed.

2 Problem Description

The end customer demand and the supplied raw material quantity in the proposed DLSPDRM are deterministic. In period t , products are replenished to downstream members, and t is called a replenishment period. The critical decision point in the proposed problem is how to decide the replenishment period and quantity. In this paper, the supply chain included an upstream member and a downstream member. Decision variable in the proposed problem is replenishment quantity y_t . Please refer to appendix A for DLSPDRM notations. The main assumptions are as follows:

- (1) $h_{2t} \leq h_{1t}$, for $t = 1, 2, \dots, T$.
- (2) $K_{1t} \geq K_{1,t+1}$, $p_{2t} \geq p_{2,t+1}$, $p_{1t} \geq p_{1,t+1}$, $s_{1t} \geq h_{1t}$, for $t = 1, 2, \dots, T$.
- (3) d_t and b_t are deterministic and uncontrollable, for $t = 1, 2, \dots, T$.
- (4) Backordering is not allowed at the upstream member, but is allowed at the downstream member.
- (5) $\alpha_2 \geq \alpha_1$.

The total cost include the upstream member's cost, cost of providing replenishment for the downstream member, and downstream member's cost, as shown in equation (1). The formulation of the DLSPDRM model is as follows:

Min

$$TC(y_t) = \sum_{t=1}^T \left[p_{2t} b_t + h_{2t} I_{2t} + K_{1t} \delta(y_t) + p_{1t} y_t + h_{1t} (I_{1t})^+ + s_{1t} (I_{1t})^- \right] \quad (1)$$

Subject to:

$$(1 - \alpha_2) I_{2,t-1} + b_t - y_t = I_{2t} \quad t = 1, 2, \dots, T \quad (2)$$

$$(1 - \alpha_1) (I_{1,t-1})^+ - (I_{1,t-1})^- + y_t - d_t = I_{1t} \quad t = 1, 2, \dots, T \quad (3)$$

$$I_{2,0} = I_{1,0} = 0 \quad t = 1, 2, \dots, T \quad (4)$$

$$y_t, I_{2t} \geq 0 \quad t = 1, 2, \dots, T \quad (5)$$

where $\delta(x) = 1$ if $x > 0$ and 0 otherwise. $(x)^+ = \max\{x, 0\}$. $(x)^- = -\min\{x, 0\}$.

Equation (2) and (3) are the inventory balance constraints at the upstream member and at the downstream member respectively. Equation (4) shows the assumption of this paper where the inventory levels is set at the level of the beginning of the initial period. Equation (5) is a non-negative integer constraint.

3 Ant Colony Optimization for DLSPDRM

3.1 The Property and Theorem of the DLSPDRM

When the demand of a period t is being met by more than one replenishment period, this is called a *demand split*.

Property 1. In the proposed problem where the quantity of supply is limited and the deterioration rate considered, there exists an optimal solution in which demand may be split.

Proof. This paper uses an example for illustration. Consider a $T = 3$ problem. Its parameters are: $h_{2t} = 1, h_{1t} = 2, K_{1t} = 3, p_{2t} = 1, p_{1t} = 1$, and $s_{1t} = 8$, for $t = 1, 2, 3$. $b_1 = 0, b_2 = 4, b_3 = 1, d_1 = 0, d_2 = 0, d_3 = 3, \alpha_2 = 0.7, \alpha_1 = 0.5$. The solution of this problem would be $y_1 = 0, y_2 = 4$, and $y_3 = 1$. Under this solution, the demand d_3 is being split; that is, the demand of period 3 is being replenished in period 2 and 3.

Property 1 shows that with the limit in supply, the solution to this problem may require the demand to be split. For example, the possible values of y_1 are the real number values of range $[0, \min\{b_1, \sum_{t=1}^T (1 - \alpha_1)^{-t+1} d_t\}]$. Under such a circumstance, the number of the possible replenishment planning of each period is enormous. As a result, to more efficiently solve the problem, the *possible partition points* (i.e., decision points) of the optimal replenishment policy have to be lessened.

Property 2. There exists an optimal solution in replenishing the demand for several sequences of periods. Demand may be split during only the first and the last of those periods. Further, when the demand is a non-negative integer, the split of this demand integer in real value only occurs in the first and the last of those periods, and is influenced by the constraints in the quantity of supply.

An equation for replenishment quantity can be derived from property 2. There exists an optimal solution so y_V is as follows:

$$y_V = (d_{V-C} - u_1) + \sum_{v=V-C+1}^V d_v + \sum_{v=V+1}^{V+L-1} (1-\alpha_1)^{-(v-V)} d_v + (1-\alpha_1)^{-L} u_2, \quad (6)$$

where V is a replenishment period, $0 < u_1 \leq d_{V-C}$, $0 < u_2 \leq d_{V+L}$. Respectively, u_1 and u_2 represent the amount of the previous replenishment period and the amount of the current replenishment period V that satisfies the last demand of the series of demands.

Theorem 1. If a lot-sizing problem considers deterioration of raw material and the limited quantity of supply, then it is NP-hard.

Proof. This proof mainly shows the decision problem of this problem can be transformed from a well-known NP-complete problem, the subset sum problem [15]. The following considers the decision problem and the subset sum problem.

- Instance: Finite set N , size $s(t) \in$ positive integer for $\forall t \in N = \{1, 2, \dots, T\}$, positive integer W .
- Question: Is there a subset $N' \subseteq N$ such that the sum of the sizes of the elements in N' is exactly W ?

Proposed problem

- Instance: make an instance for the proposed problem as follows:
 $d_1, d_2, \dots, d_{T-1} = 0; d_T = W; \alpha_2 = 1; \alpha_1 = 0; p_{2t} = 0; p_{1t} = (b_t - A)/b_t$ (note that $b_t > A$); $b_t = s(t); K_{1t} = A; h_{2t} = 0; h_{1t} = 0; s_t = \infty$.
- Question: Is there a project plan with total cost exactly equal W ?

First, the transformation is polynomial. Since the shortage cost equals $\infty > W$, the demand needs to be satisfied. In other words, if the total replenishment amount equals W (i.e., $\sum_{t \in \{1, 2, \dots, T\}} y_t = W$), to satisfy the demand, the total cost would have to equal to W . Therefore, the solution to the decision problem does exist. The discussion is shows as follows.

For $\forall t \in N = \{1, 2, \dots, T\}$:

$$G(y_t) = \begin{cases} K_{1t} \delta(y_t) + p_{1t} y_t = y_t & \text{for } y_t \in \{0, b_t\} \\ K_{1t} \delta(y_t) + p_{1t} y_t > y_t & \text{otherwise} \end{cases},$$

and therefore, the total cost of replenishment is at least W , as shown here:

$$\sum_{t=1}^T G(y_t) = \sum_{t=1}^T K_{1t} \delta(y_t) + p_{1t} y_t \geq W.$$

So, there is a total cost exactly equal to W if and only if $y_t \in \{0, b_t\}$ for $\forall t \in N$, i.e., there exists a subset $N' \subseteq N$ so $\sum_{t \in N'} b_t = W$.

Second, the optimization problem of the proposed is not NP. Therefore, the proposed problem is an NP-hard problem.

3.2 Development of Ant Colony Optimization

ACO was first introduced in Dorigo's [16] doctoral thesis. It is a meta-heuristic technique that mimics the behavior of real ants. A 0-1 binary-coded system is proposed by the properties of DLSPDRM. Algorithms developed in this paper are mainly extended from the ACO proposed by Hiroyasu *et al.* [17]. In the encoding process, the vector of a replenishment period is first produced, $\mathbf{R} = \{R_t \in \{0, 1\} \mid t = 1, 2, \dots, T\}$. In the vector, the ant then decides whether the bit should be 0 or 1 to decide the replenishment period. Next, a vector is produced $\mathbf{E} = \{E_{d'_t} \in \{0, 1\}\}$ to show whether the partition point occurs in demand d'_t , where d'_t is the demand amassed into period t to show the feasible partition point. Besides, d'_t may include a *sub-vector*, whose existence is to decide the demand split. Different solutions may have different sub-vectors, because the available quantity of supply in period t may be affected by the replenished quantity in the previous replenishment period q . In decoding, the replenishment period can be known through vector $\mathbf{R}$, and the complementary equation (6) of the partition point in demand $\mathbf{E}$ can be used to calculate the corresponding replenishment amount.

Figure 1 shows a coding example whose solution is $y_t = \{y_1 = 3.5, y_2 = 0, y_3 = 15.5, y_4 = 0, y_5 = 0\}$. In coding, $R_1 = 1$ and $R_3 = 1$ shows that period 1 and 3 are replenishment periods. In d'_1 , there is a sub-vector $\{3.5, 4\}$, and $E_{3.5} = 1$. This means the first partition point in demand is 3.5 and its demand is being met by period 1. The second partition point is E_{13} and its demand is being met by period 3. Moreover, equation (6) can be used to decode the solution.

In solving the DLSPDRM, there are two phases in the ant's construction of a solution. The first phase decides the replenishment period, whereas the second phase decides the partition point in demand given the already determined replenishment period. The partition point must be feasible (i.e., satisfy the constraint in supply). The transition probability is the main function used by the ant in its decision-making. An ant completes a tour by deciding the value of all bits through the pheromone trail, and prefers choosing the bits with the greater pheromone trail to form the solution.

Solution:

t	1	2	3	4	5
y_t	3.5	0	15.5	0	0
d_t	4	2	3	3	1

Encode:

t	1	2	3	4	5
R_t	1	0	1	0	0
d'_t	{3.5, 4}	6	9	12	13
$E_{d'_t}$	{1, 0}	0	0	0	1

Decode:

$$y_1 = 0 + 0 + 0 + 3.5 = 3.5$$

$$(C = 0, V = 1, L = 0, u_1 = 0, \text{and } u_2 = 3.5)$$

$$y_3 = 0.5 + 2 + 3 + (1 - 0.5)^{-1} 3 + (1 - 0.5)^{-2} 1 = 15.5$$

$$(C = 2, V = 3, L = 2, u_1 = 3.5, \text{and } u_2 = 1)$$

Fig. 1. An example of solution representation

The transition probability of the ant in phase 1 is as follows:

$$P^1(t) = \begin{cases} \frac{\tau_1(t)}{\tau_1(t) + \tau_0(t)} & \text{demands have not been satisfied} \\ 0 & \text{otherwise} \end{cases}, \quad (7)$$

where $P^1(t)$ is the probability of the ant choosing 1 in period t (i.e., the probability of period t being a replenishment period). $\tau_j(t)$ is the amount of pheromone trail in the position j during period t , $j \in \{0, 1\}$.

The transition probability of the ant in phase 2 is as follows:

$$P_{s^1(\tilde{d}'_t, \tilde{i})s^2}^{2k}(d'_t, i) = \frac{\tau_1(d'_t, i)}{\sum_{d''} \sum_{i''} \tau_1(d'', i'')} \quad (d'', i'') \in N_{s^1(\tilde{d}'_t, \tilde{i})s^2}^k, \quad (8)$$

where $P_{s^1(\tilde{d}'_t, \tilde{i})s^2}^{2k}(d'_t, i)$ is the probability of ant k choosing the partition point in demand (d'_t, i) during the current replenishment period s^2 when the partition point in the previous replenishment period is s^1 and the chosen partition point is $(\tilde{d}'_t, \tilde{i})$. $\tau_1(d'_t, i)$ is the amount of pheromone trail in the element i of the demand d'_t in the

sub-vector. $N_{s^1(\tilde{d}'_t, \tilde{i})s^2}^k$ is set of the feasible *applicable* partition point of demand of ant k in the current replenishment period s^2 when the previous replenishment period is s^1 and the partition point is $(\tilde{d}'_t, \tilde{i})$.

The procedure of an ant constructing a solution is as follows.

Step 1. R_t is determined through equation (7).

Step 2. Let $t = 1$.

Step 3. If $R_t = 1$, then perform the decision of the partition point (to step 3.1).

Step 3.1. With the concept of equation (6), determine $N_{s^1(\tilde{d}'_t, \tilde{i})s^2}^k$, for $s^2 = t$. Here, if the limit of the feasible partition point is a demand split, then the generation of a sub-vector is needed.

Step 3.2. Use equation (8) to determine the partition point of the current replenishment period s^2 .

Step 4. If $t = T$, end. Otherwise, next step.

Step 5. $t = t + 1$ and go to the step 3.

In the ACO algorithm, the pheromone information is crucial for the ant to find the optimal solution. ACO is applied to DLSPDRM, and ant k sets pheromone in the path of a completed tour in vector $\mathbf{R}$ as follows:

$$\Delta \tau_j^k(t) = \begin{cases} \frac{Q}{f_k} & \text{if } j \in T^{1k} \\ 0 & \text{otherwise} \end{cases}, \quad (9)$$

where Q is an adjustable parameter. f_k is the objective function value of ant k . T^{1k} is the path of a completed tour by ant k in vector $\mathbf{R}$. The traditional pheromone placement method is used in the ACO algorithm of Hiroyasu *et al.* [17].

Update the amount of pheromone trail $\tau_j(t)$, as shown in equation (10):

$$\tau_j(t) = \rho \tau_j(t) + \Delta \tau_j(t), \quad (10)$$

where ρ is a parameter between 0 and 1. $\Delta \tau_j(t) = \sum_{k=1}^U \Delta \tau_j^k(t)$.

Besides, ant k sets pheromone in the path of a completed tour in vector $\mathbf{E}$ as the following equation:

$$\Delta \tau^k(d'_t, i) = \begin{cases} \frac{Q}{f_k} & \text{if } (d'_t, i) \in T^{2k} \\ 0 & \text{otherwise} \end{cases}, \quad (11)$$

where T^{2k} is the path of a completed tour by ant k in vector $\mathbf{E}$.

Update the amount of pheromone trail $\tau_1(d'_i, i)$ as follows:

$$\tau(d'_i, i) = \rho\tau(d'_i, i) + \Delta\tau(d'_i, i) , \quad (12)$$

where $\Delta\tau(d'_i, i) = \sum_{k=1}^U \Delta\tau^k(d'_i, i)$.

A replenishment period has only one proper partition point for each period, and long-term memory consumes a huge amount of computer resources, resulting in less efficient operation, the sub-vectors added for the demands in the algorithm (i.e., the feasible partition point) should not be saved. This paper proposes two policies about developing the algorithm. In *policy 1*, the added demand split sub-vectors will be saved as long-term memory. In addition, to the added partition points, the starting pheromone value is the maximum of the current pheromone value. In *policy 2*, the increased split sub-sectors will be saved for only one iteration, and the pheromone value is consistent. The procedure of this ACO is shown in figure 2.

```

/*Initialization*/
Input:  $\rho$  ,  $Q$  ,  $U$  , and  $M$  ;
Generate  $U$  initial solutions;
Calculate objective function value of each
solution to obtain  $\mathbf{f} = \{f_1, f_2, \dots, f_U\}$ ;
Set  $\vartheta$  be the current best solution and the
corresponding  $f_\vartheta$ ;
Initialize pheromone trails;
/*Main loop*/
For  $m = 2$  to  $M$  do
    For  $k = 1$  to  $U$  do
        Determine  $R_i$  by equation (7);
        Determine  $E_{d'_i}$  by equation (8);

    End for
    Obtain  $\mathbf{f}$  by calculating objective
    function;
    Update pheromones by equations (9)-(12);
    Update  $\vartheta$  and the corresponding  $f_\vartheta$ ;
End for

```

Fig. 2. Pseudo-code for the proposed ACO

4 Numerical Analysis

It is shown in this paper that the computational complexity theory of DLSPDRM is an NP-hard problem. Thus, to lower the computational cost, this paper adopts the heuristic algorithm to solve the problems. The following experiment is designed to

discuss performing the two policies from the point of view that whether the increased demand partition points should be saved in the long-term memory. The DLSPDRM parameters are: $h_{2t} = 0.05, h_{1t} = 0.2, K_{1t} = 20, p_{2t} = 1, p_{1t} = 0.6, s_{1t} = 0.8, \alpha_2 = 0.2$, and $\alpha_1 = 0.1$. b_t is the discrete uniform distribution between 19 and 21, and d_t between 3 and 10. The other parameters are: $\rho = 0.85, \phi = 0.4, Q = 1, U = 70$, and $M = 300$. Table 1 showed the results comparison between the ACOs that implemented policy 1 and policy 2. The result shows that adding partition point and keeping long-term memory are not beneficial for improving solution quality.

Table 1. Results comparison between policy 1 and policy 2

<i>T</i>		Policy 1	Policy 2
5	Min	176.6104	176.6104
	Avg.	178.1681	176.6104
	Max	179.2066	176.6104
10	Min	364.3911	363.2226
	Avg.	370.2814	363.2226
	Max	371.7539	363.2226
15	Min	446.9711	446.9711
	Avg.	446.9760	446.9711
	Max	446.9956	446.9711
20	Min	604.2794	602.6790
	Avg.	606.6192	604.2264
	Max	613.6810	605.9526

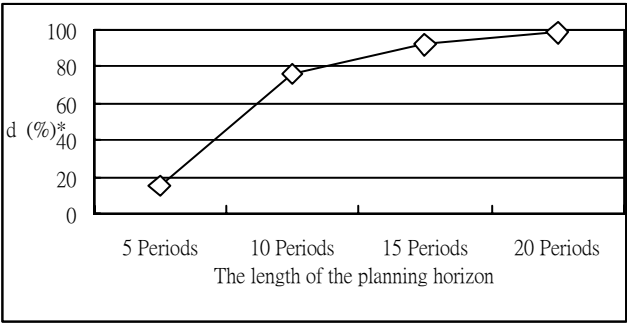


Fig. 3. The ratio of computation improvement without long-term memory. Note *: $\delta\% = (\text{Time of policy 1} - \text{Time of policy 2}) / \text{Time of policy 1} \times 100$.

Besides, figure 3 showed the calculating time improvement ratio for the algorithm without long-term memory. The result showed that the calculating time of algorithms that adopted policy 2 reduced significantly. When length of the planning horizon is larger than 15, calculating time improved for more than 80%.

5 Conclusions

Members in supply chains must be serious about the relationships between each other, and treat total cost reducing as their goal. The upstream enterprises' proper and timely replenishment to downstream enterprises is the issue that decision makers should be aware about. This paper constructed a dynamic lot-sizing problem that considered deteriorating raw materials. The feature of this proposed problem is that products' deterioration rate lowers after processing, and the critical point of this DLSPDRM is how to decide proper replenish quantity before raw materials' excessive deterioration. This paper discussed some properties of the DLSPDRM, and proved that the DLSPDRM computational complexity is NP-hard. ACO is used to solve the DLSPDRM. In the algorithm's design, policy 1 assumed that the decision points are long-term memorized, and for policy 2 only one iteration is memorized for new decision points. The test result showed that a long-term memorized decision point is not beneficial for improving solution quality, and the solving time will increase. Future research may discuss more properties for DLSPDRM to improve quality of the solving, or propose an algorithm that is more effective.

References

1. Lee, H.L., Padmanabhan, V., Whang, S.: The bullwhip effect in supply chains. *Sloan Management Review* 38(3), 93–102 (1997a)
2. Lee, H.L., Padmanabhan, V., Whang, S.: Information distortion in a supply chain: the bullwhip effect. *Management Science* 43(4), 546–558 (1997b)
3. Chen, F., Drezner, Z., Ryan, J.K., Simchi-Levi, D.: Quantifying the bullwhip effect in a simple supply chain: the impact of forecasting, lead times, and information. *Management Science* 46(3), 436–443 (2000)
4. Rogers, J.: A computational approach to the economic lot scheduling problem. *Management Science* 4(3), 264–291 (1958)
5. Moon, I., Silver, E.A., Choi, S.: Hybrid genetic algorithm for the economic lot-scheduling problem. *International Journal of Production Research* 40(4), 809–824 (2002)
6. Jensen, M.T., Khouja, M.: An optimal polynomial time algorithm for the common cycle economic lot and delivery scheduling problem. *European Journal of Operational Research* 156(2), 305–311 (2004)
7. Wagner, H.M., Whitin, T.M.: Dynamic version of the economic lot size model. *Management Science* 5(1), 89–96 (1958)
8. Lee, C.Y., Cetinkaya, S., Jaruphongsa, W.: A dynamic model for inventory lot sizing and outbound shipment scheduling at a third-party warehouse. *Operations Research* 51(5), 735–747 (2003)
9. Jaruphongsa, W., Cetinkaya, S., Lee, C.Y.: Warehouse space capacity and delivery time window considerations in dynamic lot-sizing for a simple supply chain. *International Journal of Production Economics* 92(2), 169–180 (2004)

10. Özdamar, L., Birbil, S.I.: Hybrid heuristics for the capacitated lot sizing and loading problem with setup times and overtime decisions. *European Journal of Operational Research* 110(3), 525–547 (1998)
11. Smith, L.A.: Simultaneous inventory and pricing decisions for perishable commodities with price fluctuation constraints. *INFOR*. 13(1), 82–87 (1975)
12. Friedman, Y., Hoch, Y.: A dynamic lot-size model with inventory deterioration. *INFOR*. 16(2), 183–188 (1978)
13. Hsu, V.N.: Dynamic economic lot size model with perishable inventory. *Management Science* 46(8), 1159–1169 (2000)
14. Hsu, V.N.: An economic lot size model for perishable products with age-dependent inventory and backorder costs. *IIE Transactions* 35(8), 775–780 (2003)
15. Garey, M.R., Johnson, D.S.: *Computer and Intractability-A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York (1979)
16. Dorigo, M.: *Optimization, learning and natural algorithms*. Ph.D thesis, Politecnico di Milano, Italy (1992)
17. Hiroyasu, T., Miki, M., Ono, Y., Minami, Y.: *Ant colony for continuous functions*. The Science and Engineering, Doshisha University, Japan (2000)

Appendix: Notation of DLSPDRM Is Defined as Follows

Notations

t the period index

T the planning horizon

d_t the demand at the downstream member in period t

b_t the procurement quantity of material at the upstream member in period t

y_t the production and transportation (i.e., replenishment) quantity of product in period t

I_{2t} the inventory level of material at the upstream member at the end of period t

I_{1t} the inventory level of product at the downstream member at the end of period t

K_{1t} the fixed cost of production and transportation to the downstream member in period t

p_{2t} the unit procurement cost of material at the upstream member in period t

p_{1t} the unit cost of production and transportation to the downstream member in period t

h_{2t} the unit holding cost of material at the upstream member in period t

h_{1t} the unit holding cost of product at the downstream member in period t

s_{1t} the unit shortage cost of product at the downstream member in period t

α_2 the inventory deterioration rate of raw material at the upstream member

α_1 the inventory deterioration rate of product at the downstream member

An Algorithm for Propagating-Impact Analysis of Process Evolutions

Jeewani Anupama Ginige and Athula Ginige

University of Western Sydney, School of Computing and Mathematics,
Locked Bag 1797, Penrith South DC NSW 1797, Australia
{j.ginige,a.ginige}@uws.edu.au

Abstract. Business processes evolve due to different reasons. Evolution of business processes essentially means changing its process elements namely: actions, participants, and process objects; which are associated to each other in various ways. In the event of one process element change, the above-mentioned associations create propagating-impact. Therefore in process evolution management, it is imperative to have a business process modelling tool that can completely and cohesively capture associations among process elements. In our previous research [1] we have developed such a process modelling tool using Kleene Algebra with Tests - KAT [2]. In this paper, we present an algorithm that facilitates locating the propagating-impact, of a process element change, across the entire process. The proposed mechanism initially, maps the KAT expression of a process, into a binary-tree structure. Then using this binary-tree, the created propagating-impact is extracted under four categories as Direct, Indirect, Secondary and Non-cautionary (DISN) impacts [1].

Keywords: Process Evolution, Impact Analysis, Propagating-impact, Process Modelling, Algebraic Modelling, Kleene Algebra with Tests.

1 Introduction

Evolution of a business process refers to changing process elements: actions, participants, and process objects. *Actions* refer to the tasks that need to be carried out in achieving the goals of the business process. *Participants* could be people, machinery, software, or any other form of resource that is assigned to carryout process actions. *Object* of a process refers to either a physical object, informational object, or a combination of both on which actions are performed.

In business processes, above process elements are associated to each other in two ways. Firstly, the *Dynamic Associations* show the correlations among actions namely: Sequence, Parallel, Conditional Choice, Synchronisation, and Simple Merge. The *Static Associations* denote the all types of correlation among other process elements. An example for a static association between participants and actions is, in a particular process any given participant can be obliged, permitted or prohibited to perform process actions [3].

Irrespective of whether the associations are dynamic or static, these set the foundation for *propagating-impact* in the event of a process element change. This outlines

the imperativeness of having a business process modelling tool that can completely and cohesively capture all types of associations among process elements.

In our previous research [1] we have developed such a process modelling tool, that can completely and cohesively capture both dynamic and static associations among process elements. This process modelling tool is based on Kleene Algebra with Tests - KAT [2].

Now the question remains, *how can we accurately locate the propagating-impact that gets created across the entire process as a result of a process element change?* Such accurate identification of propagating-impact leads to better management of process evolutions.

Workflow analysis is a term closely associated with management of process evolutions. Workflow analysis constitutes three analytical concepts: Validation, Verification, and Performance Analysis [4]. Validation refers to assuring the modelled process is exactly what is required by the business and is carried out by means of process simulation [5, 6]. Verification refers to ensuring the structural and syntactic correctness of process models according to the process modelling notations or language used [7-10]. Performance Analysis refers to understanding any bottlenecks or any other performance issues using different approaches [11, 12]. Workflow analysis leads to refinement of process models or further redesign of the process in order to rectify any identified errors or inconsistencies.

The term *evolution impact analysis* is different to the term workflow analysis [4]. The evolution impact analysis denotes the assessment of risks and impacts of changes prior to them being introduced to process models [13]. In some sense, business process evolution impact analysis and workflow validation have the same end goal. This end goal is to assure that the modelled process is what is required by the organisation. However, the major difference between these two is in approaches and techniques used. Workflow validation mainly uses simulation for assuring the consistency of the process. Simulation greatly relies on the human's ability to recognise any differences between the required and modelled process and is only as good as the data set that is used for the simulation purposes [14]. In contrast, methodologies used in an evolution impact analysis, identify the risks of changes, with the objective of minimising subsequent errors and inconsistencies. When proper tools are used for this impact analysis task, it can be assured that the modelled process is free of human errors.

Change impact analysis is highly researched and practised in the software engineering domain [13, 15]. For example, the mini-cycle of change as described by Zhao et al. [15] and Yau et al. [16] indicates the need for a planning phase consisting of program comprehension and change impact analysis. Similarly, process impact analysis gives the opportunity for comprehension of already automated process and finds the propagating-impact.

Process evolution analysis is a largely neglected area in workflow research [13]. This gap has been identified and recently some work has been proposed in this area [13, 17, 18]. While these studies attempt to address some issues of impact analysis, still problems exist in this domain. In particular, most previous research fails to provide a holistic solution, which can manage evolution of process actions, participants, and business objects.

In this paper, we present an algorithm that facilitate in locating the propagating-impact of one process element evolutions into the rest of the process. In addition, the

located propagating-impact is classified into four categories as Direct, Indirect, Secondary and Non-cautionary (DISN) impacts (further clarified in section 3 of this paper). In finding DISN impact, the linear process expressions based on KAT are used. The proposed mechanism initially, maps the KAT expression of a process, into a binary tree structure. Then by traversing the binary tree, propagating-impact on the rest of the process is located. Further the proposed algorithm is validated based on a hypothetical business process (leave approval process) and a possible evolution scenario associated with it.

2 Background

In this section we briefly present the background information that is vital for the core contributions of this paper.

2.1 KAT for Process Modelling

There are a number of formal process-modelling mechanisms to capture the inner workings of a process. These formal mechanisms include Petri-Nets [19-21], Process Algebra [22-24] and Kleene Algebra with Tests–KAT [2, 25]. In our previous research [1] , though we had demonstrated the use of KAT for process modelling, there we did not justify the usage of KAT over other formalisms. Therefore here firstly the suitability of KAT over other formalism is further established.

A formalism used for modelling of business process needs to facilitate a) the Ability to capture both *Dynamic and Static Associations* and b) Ability to represent the entire process in a single linear expression that allows computer manipulation. Hence, the above a) and b) are used as the evaluation criteria in finding a suitable formalism for process modeling (Table 1).

Table 1 presents a comparison between Petri-Nets, Process Algebra, and KAT based on their ability to achieve the afore-mentioned requirements. The symbols used indicate the degree of support: (+) indicates that the formalism fully supports the

Table 1. Compariosn between Petri-Nets, Process Algebra and KAT in cohisively and completely capturing various associations among process elements

Formalism	Dynamic Associations					Static Associations	Ability to create linear process expressions
	Sequence	Parallel	Choice	Synchronisation	Merge		
Petri-Nets	+	+	+	+	+	+	- [26]
Process Algebra	+	+	+	- [26]	+	- [26]	+
KAT	+	>	+	+	+	+	+ [27]

requirements without any modifications, (-) indicates the inability to support the requirement and (>) indicate the ability to support with certain known extensions or minor modifications.

Although Petri-Nets based models are appropriate for creating graphical process notations, according to the findings presented in Table 1; in Petri-Nets it is not able to create linear process expressions. Using Process Algebra it is possible to create linear expressions of processes. But Process Algebra fails to capture certain synchronisations and static associations in an efficient manner. This is mainly due to the uni-partite (single type of element) nature of Process Algebra [26].

KAT is a two-sorted (bi-partite) algebraic structure $(B, K, +, \cdot, *, 0, 1, \sim)$, where B is a subset in K . $\sim$ is a unary operator (similar to negation) defined only on B . $*$ is also a unary operator is similar to iteration defined only on elements from K . 0 and 1 are special elements that symbolise null and skip actions [2]. KAT being a bi-partite algebra it is able to capture both dynamic and static associations. Further, it is possible to create linear expressions that can be analysed as required by this research. In addition, KAT has special elements such as 0 and 1 that represents null and skip actions respectively. In this view, KAT stands out to be the most suitable formalism to capture associations among process elements.

However, a disadvantage of KAT over Process Algebra is that it lacks a direct operator to represent parallelism. Therefore in our previous research [1] we have exemplified the usage of choice (+) and iteration (*) operators in combination, to represent the parallel flow control.

2.2 Process Example Using KAT

Here a simple process example is presented to re-capture the usage of KAT in process modelling. This example is further used for the validation of the main research findings of this paper.

Figure 1 shows a hypothetical process example of a typical leave approval process of an organisation.

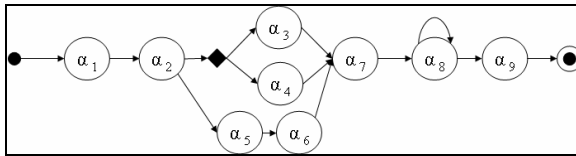


Fig. 1. Hypothetical Leave Approval Process

The actions (alphas) associated with this leave approval process depicted in Figure 1, are as follows:

- α_1 – The applicant fills in the leave approval form and forward to the *immediate manager*
- α_2 – *Immediate manager* checks the leave approval form

- α_3 - If the leave is for annual leave the *quality officer* approves the leave form
- α_4 - If the leave is for sick leave the *line manager* approves it
- α_5 - Alternate staffing is arranged by the *line manager*
- α_6 - Alternate staffing is checked by the *quality officer*
- α_7 - *Departmental Manager* checks the leave form and alternate staffing arrangements
- α_8 - *Shift planners* re-schedule the shift work until it allows them to meet production targets
- α_9 - Final approval for leave is given by the *production manager*

This ‘leave approval process’, has the above nine process actions (α_1 to α_9) and following are the guard elements (phis) associated with this process, based on the definitions introduced in our previous work [1]:

Table 2. List of Guard Element (Phi) Notations used in the sample process

Phi	KAT definition	Comments
ϕ_1	P (applicant, α_1)	Applicant is <i>permitted</i> to perform action α_1
ϕ_2	V((DATA=>{defines the fields in the leave form })), α_1)	Identifies the information fields required to be collected in the leave application form
ϕ_3	O(immediate manager, α_2)	Immediate manager is <i>obligated</i> to perform action α_2
ϕ_4	PC(applicant, immediate manager (characteristic =>{ report to}))	Defines the mandatory relationship between roles ‘applicant’ and ‘immediate manager’
ϕ_5	P(quality officer, α_3)	Quality officer is <i>permitted</i> to perform the action α_3 .
ϕ_6	P(manager, α_4)	Manager is <i>permitted</i> to perform the action α_4
ϕ_7	P(line manager, α_5)	Line manager is <i>obligated</i> to perform the action α_5
ϕ_8	O(quality officer, α_6)	Quality officer is <i>obligated</i> to perform the action α_6 .
ϕ_9	O(departmental manager, α_7)	Departmental manager is <i>obligated</i> to perform action α_7
ϕ_{10}	O(shift planners, α_8)	Shift planners are obligated to perform action α_8
ϕ_{11}	O(production manager, α_9)	Production manager is <i>obligated</i> to perform action α_9
ϕ_m		This is an internal condition used to manage the parallel processing
ϕ_n		This is an internal condition used to manage the iteration of shift planning activity that checks production can meet targets

The KAT expression E below captures the process given in Figure 1 using the above described alphas and phis and the flow controls based on the axioms of KAT.

$$E = (\phi_1 \phi_2 \alpha_1) (\phi_3 \phi_4 \alpha_2) \phi_m ((\phi_5 \alpha_3 + \phi_6 \alpha_4) + (\phi_7 \alpha_5) (\phi_8 \alpha_6))^* (\phi_9 \alpha_7) \phi_n (\phi_{10} \alpha_8)^* (\phi_{11} \alpha_9) \quad (1)$$

The above KAT expression (1) will be further utilised in later sections in validating the algorithm presented in this paper.

3 DISN Impacts of Process Element Evolution

This notation of evolutions impact analysis and impact categorisation in relation to business processes is not discussed in detail in previous work. The research that acknowledges the need for impact analysis is presented by Soffer [18], Bodhuin et al. [13], and Min, Bae, Cho, and Nam [28]. Among these works only Soffer [18] offers a categorisation of impacts into two types as global or local. Here we provide a detailed classification to represent different kinds of propagating-impacts. This classification recognises four categories named Direct, Indirect, Secondary, and Non-cautionary impacts (DISN impacts) as follows:

- **Direct** impact refers to particular guard elements or actions in a process that is directly affected due some change— E.g. removal of the organisational role quality officer.
- **Indirect** impact denotes particular guard elements or actions in a process, which cannot be computed or executed, due to the direct impact on some other elements— E.g. if quality officer position is removed, the actions that are (α_3 and α_6 in Figure 1) to be performed by this role has an indirect impact.
- **Secondary** impact indicates particular guard elements or actions that can be computed or executed, but cannot be merged with the rest of the process for successful completion— E.g. If the quality officer is removed the while actions α_4 and α_5 can be performed, they cannot merge at α_7 as α_3 and α_6 has indirect impacts.
- **Non-cautionary** impact refers to the guard elements and actions that logically appear before the point of change or in a parallel branch, which has no or minimal impact.

4 Mapping of Process Expressions in KAT to Binary Tree Structures

As demonstrated in section 2.2 above, KAT expressions cohesively and completely capture the associations among process elements. However, for further manipulation and analysis of these expressions, it required to represent these KAT expressions in a suitable data structure.

Tree structures are considered efficient in representing complex and linear structures [29], similar to the ones in KAT expressions. In particular, binary trees are advocated to be used due to its regularity of the elements, which involves just three elements – left link, data, and right link. In addition, Shave [29] demonstrates a mechanism to convert any general multi-branched tree structure into a binary tree structure, while preserving its structure.

There are five types of main flow constructs present in KAT expressions. These are: sequential flow of actions, iteration of an action, action guarded by a negated condition, conditional choice between actions and parallelism between actions. Figure 2 demonstrates the representation of these five types of flow control patterns in binary tree structures.

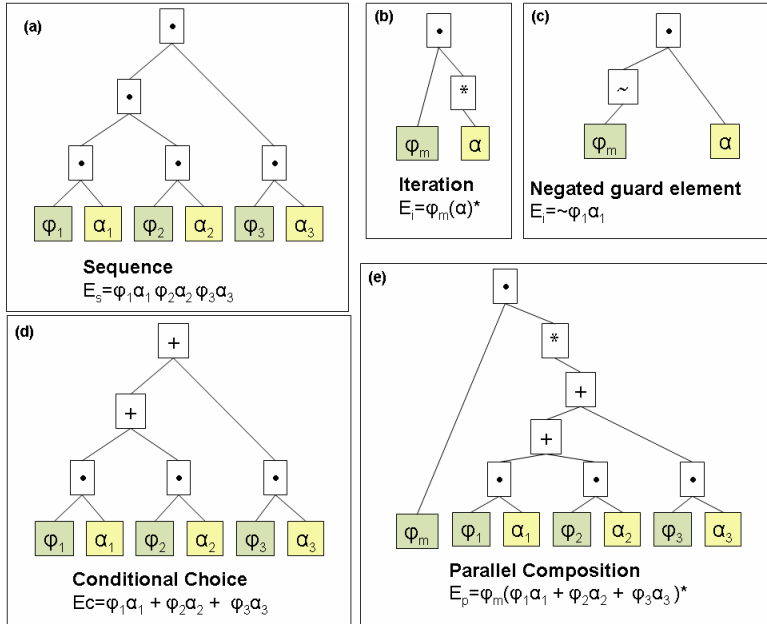


Fig. 2. Binary Tree representation of main flow control patterns in KAT

When creating and manipulating KAT expressions in binary tree structures the following structural invariant needs to be maintained.

- There can be up to a maximum of two branches hanging from a particular node.
- Phis and alphas always need to be in leaf nodes only.
- The guard element (a phi) of a particular action (an alpha) always needs to be in the left branch of the parent node.
- When the parent node holds a plus (+) operator as the node value, there has to be two children elements and they can be:
 - either two phis or two alphas - the combination of a phi and alpha cannot be hanging from a plus (+) operator parent node, or
 - a combination of any operators pluses (+), dots (•), stars (*) and negation (~).
- When the parent node holds a dot (•) operator as the node value, there has to be two children elements and they can be a combination of any phis, alphas or any operator pluses (+), dots (•), stars (*) and negation (~).
- When the parent node holds a star (*) operator as the node value, there can be only one child node (as this is a unary operator) and the child element can be either an alpha, a plus (+) or a dot (•).
- When the parent node holds a negation (~) operator as the node value, there can be only one child node (as this is a unary operator) and the child element can be either a phi, a plus (+) or a dot (•).

The writing of a KAT expression into a binary-tree (or recreation of KAT expressions from an existing binary-tree) can be done using pre-order traversal mechanism or sometimes known as symmetric order [30]. This method guarantees that it will visit every element of the tree once and once only [29]. In addition, it preserves the sequential ordering of actions when transforming to and from KAT expressions and tree structure.

Figure 3 below shows the binary tree representation of sample KAT expression (1) (introduced in section 2.2) based on the control patterns given in Figure 2 and the set of invariants explained above.

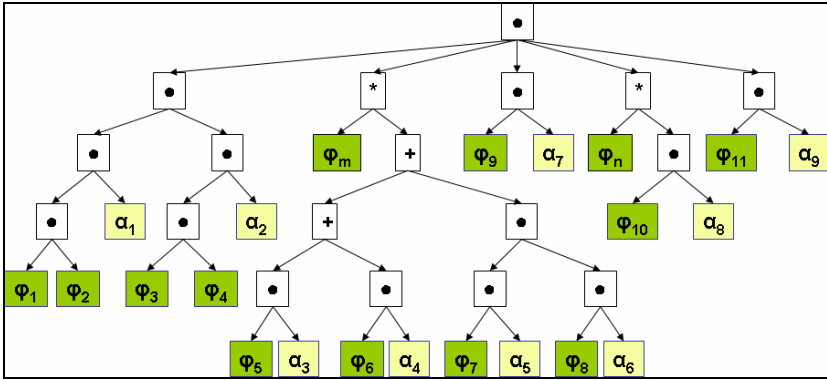


Fig. 3. Binary-tree representation of KAT expression (1)

5 Algorithm in Locating DISN Impact of Process Evolution

The following are the algorithmic-steps (at the highest level) for searching a binary tree structure similar to Figure 3.

- **Algorithmic-Step-1:** Locate the position of directly impacted process element by traversing the binary-tree using in-order traversal method [30, 29]
- **Algorithmic-Step-2:** Traverse upward until the root is reached.
- **Algorithmic-Step-3:** In every step in this upwards traversal, phis and alphas in left and right branches of the tree are marked either as in indirect, secondary, or non-cautionary impacted elements.

In Algorithmic-Step-3 above, a number of factors are taken into consideration when deciding the type of impact. These factors include, whether the directly impacted element is a phi or an alpha, whether this changing element is on left or right side branch of the tree, the type of the notation (phi, alpha or the type of operator plus (+), dot (.), star (*) or negation (~)) held in the parent node, and the type (phi or alpha) of sibling node and its children nodes. This complex logic involved in identifying the impacts on other elements is summarised in Table 3.

Table 3: Decision Table for the Algorithmic-Step-3, which locates D-Direct, I-Indirect, S-Secondary and N-Non-cautionary impact of a process element evolution

IN EACH STEP BACK UP TO THE ROOT (ALGORITHMIC-STEP-2)			When Direct Impacted Node is a Phi			When Direct impacted node is an Alpha		
Parent	Sibling	Sibling	LEFT Brach of the parent node – Impact on		RIGHT Brach of the Parent Node – Impact on	LEFT Brach of the parent node Impact on		RIGHT Brach of the Parent Node – Impact on
			Sibling	Sibling's Children		Sibling	Sibling's Children	
Dot (•)	alpha (α)	-	-	-	-	I	-	N
	phi (φ)	S	-	-	-	-	-	I
	Dot (•)	-	-	N	-	-	N	-
	Plus(+)	-	-	N	-	-	N	-
	Star (*)	-	-	N	-	-	Alpha-N	-
	Negate(~)	-	-	Phi-N	-	-	Phi-N	-
Plus(+)	alpha (α)	-	-	-	-	N	-	N
	phi (φ)	N	-	-	-	-	-	-
	Dot (•)	-	-	Phi-N	-	-	N	-
	Plus(+)	-	-	Phi-N	-	-	N	-
	Star (*)	-	-	-	-	-	Alpha-N	-
	Negate(~)	-	-	Phi-N	-	-	-	-
Star(*)		-	-	-	-	-	-	-
Negate(~)		-	-	-	-	-	-	-
phi (φ)		-	-	-	-	-	-	-
alpha (α)		-	-	-	-	-	-	-

In this table the dash (-) indicates the situations where it is logically not possible for sibling elements to exist based on the type of directly impacted node and parent node.

There are mainly three sections in Table 3. The first section refers to Algorithmic-Step-2 of the high-level algorithm, which checks the type of parent nodes and sibling nodes. The other two sections indicate the logic involved in deciding the type of impact (indirect, secondary and non-cautionary). In addition to the above-mentioned factors, it is also vital to watch that a particular constraint (ϕ) or actions (α) are not placed in two categories of impacts. The way to resolve this is by ranking the impacts. The impacts are ranked in the order of direct, indirect, secondary and non-cautionary; where direct has the highest impact and non-cautionary has the lowest impact. If an element is found to be in two categories, it is placed in the higher impact category. There are a number of places in the table which has (-) entries. These indicate the situations where it is logically not possible for sibling elements to exist based on directly impacted node and parent node.

6 Validation of the DISN Impact Locating Algorithm

In relation to the leave approval example provided previously, the following hypothetical evolution scenario of “role quality officer is removed from the organisational structure” is considered to validate the proposed algorithm. Using the a prototypic implementation of the algorithm the DISN impacts are extracted.

In this evolution scenario, the removal of role of the participant is directly affecting the guard elements ϕ_5 and ϕ_8 . Hence, it is not possible to evaluate these guard elements, due to the non-existence of the role ‘quality officer’. This is shown using label D- Direct in Figure 4. When the both guard elements ϕ_5 and ϕ_8 cannot be executed, the associated actions of those guard elements α_3 and α_6 have indirect impact on them. This is denoted using label I-Indirect in Figure 4. The label S-Secondary shows the action α_4 that actually can be executed, but is not able to merge with the rest, for a successful completion. This secondary is due to action α_4 being in parallel with the directly impacted element. The actions and guard elements that are logically prior to the directly impacted elements are denoted using, label N-Non-Cautionary.

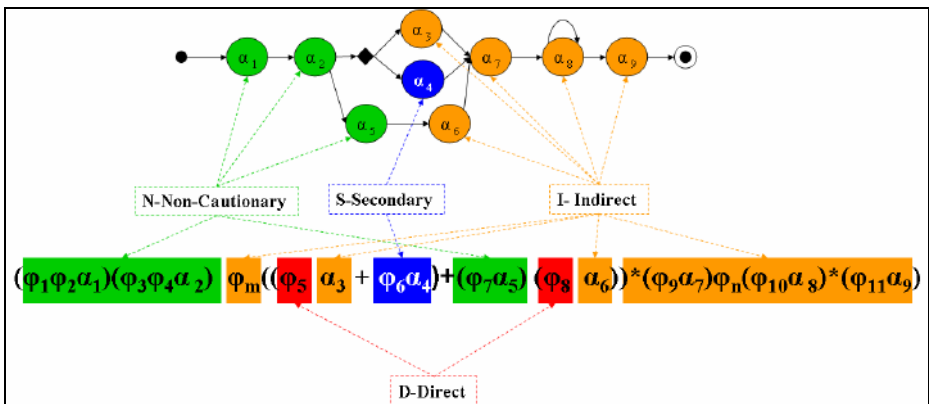


Fig. 4. DISN impact of Evolution Scenario 1

7 Conclusion

In this paper we presented an algorithm which shows the mechanism of locating propagating-impact of process evolution. Based on the prototypic implementation of this logic using C++, we have demonstrated how it can be applied in a simulated real life scenario.

This approach for process impact analysis gives the opportunity for the comprehension of the already automated processes and finding the propagating-impact. This accurate identification of propagating-impact prior to initiating the modification task facilitates the efficient execution of the task. This in return reduces the downtime of systems that support critical business processes, thus reduces the costs and effort required to manage business process evolution.

Further work that arise from this research can lead to enhancing the algorithm for the purposes of managing running instances of a process that are effected as a result of the core process definition being changed.

References

- [1] Ginige, J.A., Ginige, A., Sirinivasan, U.: KAT based CAD Model of Process Elements for Effective Management of Process Evolution. In: Workshop on Technologies for Collaborative Business Processes (TCoB-2007) in ICEIS 2007, Funchal, Madeira - Portugal (2007)
- [2] Kozen, D.: Kleene Algebra with Tests, p. 17. ACM, New York (1999)
- [3] Ginige, J.A., Ginige, A., Sirinivasan, U.: CAD model of Process Elements: Towards Effective Management of Process Evolution in Web-based Workflows. In: Sixth International Conference on Computer and Information Science, Melbourne Australia (2007)
- [4] van der Aalst, W.M.P.: Making Work Flow: On the Application of Petri nets to Business Process Management. In: Esparza, J., Lakos, C.A. (eds.) ICATPN 2002. LNCS, vol. 2360, pp. 1–22. Springer, Heidelberg (2002)
- [5] Bosilj-Vuksic, V., Jaklic, J., Popovic, A.: Business Process Change and Simulation Modelling. *Systems Integration*, 29 (2005)
- [6] van Hee, K., Oanea, O., Post, R., Somers, L., et al.: Yasper: a tool for workflow modeling and analysis. In: Sixth International Conference on Application of Concurrency to System Design (2006)
- [7] Eshuis, R., Wieringa, R.: Verification support for workflow design with UML activity graphs. In: 24th International Conference on Software Engineering - ICSE 2002, Orlando, Florida, USA (2002)
- [8] Marjanovic, O.: Dynamic Verification of Temporal Constraints in Production Workflows. In: 11th Australian Database Conference (2000)
- [9] Sivaraman, E., Kamath, M.: Verification of Business Process Designs Using Maps. In: Golden, B.L., Raghavan, S., Wasil, E.A. (eds.) *The Next Wave in Computing, Optimization, and Decision Technologies*, vol. 29, pp. 303–318. Springer, Heidelberg (2005)
- [10] van der Aalst, W.M.P.: Verification of Workflow Nets. In: Azéma, P., Balbo, G. (eds.) ICATPN 1997. LNCS, vol. 1248, pp. 407–426. Springer, Heidelberg (1997)
- [11] Aiello, R.: Workflow Performance Evaluation, University of Salerno, Italy, p. 158 (2004)

- [12] Stefanov, V., List, B.: A Performance Measurement Perspective for Event-Driven Process Chains. In: Sixteenth International Workshop on Database and Expert Systems Applications (2005)
- [13] Bodhuin, T., Esposito, R., Pacelli, C., Tortorella, M.: Impact Analysis for Supporting the Co-Evolution of Business Processes and Supporting Software Systems. In: Workshop on Business Process Modeling, Development, and Support (BPMDS), Riga, Latvia (2004)
- [14] Jansen-Vullers, M.H., Netjes, M.: Business Process Simulation-A Tool Survey. In: Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools, Aarhus, Denmark (October 2006)
- [15] Zhao, J., Yang, H., Xiang, L., Xu, B.: Change impact analysis to support architectural evolution. *Journal of Software Maintenance and Evolution Research and Practice* 14, 317–333 (2002)
- [16] Yau, S.S., Collofello, J.S., MacGregor, T.: Ripple effect analysis of software maintenance. In: IEEE Computer Society's Second International Computer Software and Applications Conference, COMPSAC 1978 (1978)
- [17] Ramesh, B., Jain, R., Nissen, M., Xu, P.: Managing context in business process management systems. *Requirements Engineering* 10, 223–237 (2005)
- [18] Soffer, P.: Scope Analysis: Identifying the Impact of Changes in Business Process Models. In: Regev, G., Soffer, P., Bider, I. (eds.) *Software Process Improvement And Practice*, vol. 10, pp. 393–402. John Wiley & Sons, Ltd, Chichester (2005)
- [19] Murata, T.: Petri nets: Properties, analysis and applications. *Proceedings of the IEEE* 77, 541–580 (1989)
- [20] Reisig, W.: Petri Nets, An Introduction. In: Brauer, G.R.W., Salomaa, A. (eds.) *Mono-graphs on Theoretical Computer Science*. Springer, Heidelberg (1985)
- [21] van der Aalst, W.M.P., van Hee, K.M., Houben, G.J.: Modelling and analysing workflow using a Petri-net based approach. In: 2nd Workshop on Computer-Supported Cooperative Work, Petri nets and related formalisms (1994)
- [22] Basten, A.A.: In Terms of Nets: System Design with Petri Nets and Process Algebra, p. 247. Eindhoven University of Technology (1998)
- [23] Fokkink, W., Zantema, H.: Basic Process Algebra with Iteration: Completeness of its Equational Axioms. *The Computer Journal* 37, 259–267 (1994)
- [24] van Glabbeek, R.J.: Bounded nondeterminism and the approximation induction principle in process algebra. In: Brandenburg, F.J., Wirsing, M., Vidal-Naquet, G. (eds.) *STACS 1987. LNCS*, vol. 247, pp. 336–347. Springer, Heidelberg (1987)
- [25] Kozen, D.: Kleene algebra with tests, *Transactions on Programming Languages and Systems*, pp. 427–443. ACM, New York (1997)
- [26] van der Aalst, W.M.P.: Pi calculus versus Petri nets: Let us eat “humble pie” rather than further inflate the “Pi hype” (2003)
- [27] Schewe, K.D., Thalheim, B.: Conceptual modelling of web information systems. *Data and Knowledge Engineering* 54, 147–188 (2005)
- [28] Min, S.Y., Bae, D.H., Cho, S.C., Nam, Y.K.: Management of Workflow over the Web Supporting Distributed Process Evolution. In: Hui, L.C.-K., Lee, D.-L. (eds.) *ICSC 1999. LNCS*, vol. 1749, pp. 367–372. Springer, Heidelberg (1999)
- [29] Shave, M.: *Data Structures*. McGraw-Hill Book Company Limited, Maidenhead (1975)
- [30] Knuth, D.E.: Sorting and Searching Algorithms. In: Varga, R.S., Harrison, M.A. (eds.) *The Art of Computer Programming*, vol. 3, p. 710. Addison-Wesley Publishing Company, Massachusetts (1973)

Business Process Improvements in Production Planning of ERP System Using Enhanced Process and Integrated Data Models

Premaratne Samaranayake

School of Management, University of Western Sydney
Locked Bag 1797, Penrith South DC NSW 1797, Australia
p.samaranayake@uws.edu.au

Abstract. This paper presents a framework for business process improvements for process integration, automation and optimisation in an ERP system environment. Main features of the proposed framework include (i) enhanced process models and integrated data models, incorporating many components and relationships/links and (ii) process optimisation through improved process logics and additional functionalities. Business processes are modeled using enhanced Event-driven Process Chain (EPC) methodology, incorporating additional logics and functionalities. Master and transaction data associated with business processes are modeled using unitary structures. Transaction data incorporate integrated master data and functional tasks such as good issue and receipt of production order cycle, for improved planning and scheduling of components. Potential improvements include simultaneous planning of many components, forward planning and finite loading of resources. The paper concludes that enhanced process and integrated data models improve functional applications and eliminate the need for separate and interfaced applications for process improvements in ERP system environment.

Keywords: Data structures, applications, integration, process improvements.

1 Introduction

Business process improvements through business process re-engineering projects and enterprise resource planning (ERP) systems have been the subject of considerable interests in many organisations. In this regard, ERP systems have emerged as the core of successful data, information and knowledge management through integrated functional applications across entire organisation. In recent times, the adoption of ERP systems are becoming more of supporting their businesses under ever changing and evolving environment of diminishing market shares, tough competition, increasing customer expectations and globalization. Despite widespread use of ERP systems, many companies are beginning to realize that the real impact of ERP systems on management styles and practices is actually well below expectations, especially on the front of organisational integration [1].

It has been recognized that Business Process Re-engineering (BPR) plays a significant role in ERP system implementations [2]; [3]. Martin and Cheung (2005) demonstrate through a case study that significant improvements through BPR can be achieved after the implementation of ERP systems. Further, process integration within ERP has a potential for bringing the maximum benefits of BPR [3]; [4]. However, it is a challenge for many organisations to carry out BPR project effectively [5]. With ERP systems, BPR has a dual role – one of result and the other as a prerequisite [2]. Moreover, BPR and ERP can be supportive of each other [6], where ERP systems can support BPR, through implementation of business process integration using applications and automation using workflows. Thus, success of ERP implementations and subsequent improvements on performance are dependent on how well benefits of BPR projects are achieved and maximized.

In recent times, research activities on business process improvements are confined to areas such as integration efforts, business process modeling, simulation, reference model for SMEs, and narrow spectrum of industries [7]; [8]; [9]; [10]; [11]; [12]; [13]; [14]; [15]; [16]; [17]. Recently, Davenport *et al.* (2004) concluded that factors most associated with achieving value from enterprise systems were integration, process optimisation, and use of enterprise systems data in decision making. However, there is limited research on those aspects to date. Beretta (2002) identified the lack of organisational integration with the business processes in ERP while McAdam and McCormack (2001) concluded that there is lack of research exploring the integration of business processes extending throughout supply chains.

On process automation and optimisation aspects, Lau *et al.* (2003) proposed a system with features of ability to reconfigure and customize business workflows of process automation, allowing the integration of workflow in a flexible way within industry environment. Samaranayake and Chan (2006) proposed a framework for integrating applications and workflows within an ERP system environment, based on enhanced Event-driven Process Chain (EPC) methodology. Further, limited literature on business process optimisation are confined to very narrow areas such as optimisation of: (i) business to business relationships [18], (ii) business process designs using mathematical model for minimizing cost and duration of the process [19], and (iii) ERP systems using roles, knowledge, better integration and user competencies [20].

In this research, the focus is business process improvements through enhanced process and integrated data models. A framework for process integration, automation and optimisation is proposed. Process optimisation is incorporated into functional applications, using improved process models and additional functionalities based on unitary structuring technique [21]. Process improvements are supported by integrated master and transaction data. Enhanced process and integrated data models are demonstrated using a numerical example within production order cycle of ERP.

This paper is organized as follows. The proposed framework for business process integration, automation and optimisation is presented first, followed by business process improvements using enhanced EPC methodology. Next, integrated master and transaction data in production planning of ERP is presented with a numerical example drawn from the industry. Finally, this paper concludes with research findings and recommendations for future research directions.

2 Framework for Business Process Integration, Automation and Optimisation

The framework for business process integration, automation and optimisation is based on enhanced process and integrated data models. Main features of the framework include (i) improved process models using enhanced EPC methodology [10], incorporating many components, relationships and links; and (ii) process optimisation with additional functionalities using integrated process/data models. Thus, the framework provides a basis for developing a set of transactions not only carrying out execution of process steps but also execution of relevant workflows and/or optimisation within the process. For example, the function “quotation to be created from inquiry” within the standard sales order process can be automated through the business workflow “quotation approval”. Further, production order process can be optimized by incorporating scheduling functions for finite loading of resources into the production order creation function, with required data and organisational elements. This would not only eliminate the need for separate execution of applications and business workflows, but also provide required functionalities for optimizing complex processes as part of functional applications rather than separate applications in ERP system environment.

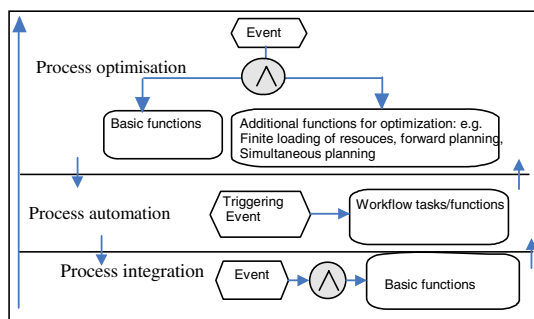


Fig. 1. Framework for process integration, automation and optimisation

The framework (Fig. 1) represents three building blocks of business processes: process integration, process automation and process optimisation. Business processes are improved at the integration and automation levels, before processes become candidates for optimisation. At the first level, process integration is represented by a combination of functions and events. Those functions are represented by combinations of many types of components (activities, resources, materials and suppliers) depending on the type of function and the purpose within the process. These components are linked together using three different relationships: parent-component (hierarchical), component-component (sequential) and activity precedence (closed-network). Similarly, process automation, supported by process integration, is represented by process components such as functions, events (triggering and terminating), resources linked with tasks and associated relationships between components.

In the proposed framework, basic functions at the integration level, functions and tasks at the automation and additional functions at the optimisation are represented by not only activities/tasks but also by associated resources, materials, suppliers, customers and relationships between components. Thus, functions with these components form the basis of enhanced process and integrated data models. Overall, business process integration, automation and optimisation are based on enhanced business process models using unitary structuring technique.

The proposed framework can be used to represent process cycles in terms of process components including workflow processes, data elements and structures involved, and relationships between components. Relationships are represented by links between components with appropriate precedence (parent-component, component-component and activity precedence). Moreover, the proposed framework provides the flexibility and maintainability of many process cycles, and can be used to simulate existing processes to achieve better planning and execution outcomes for all the components involved. The framework is also a basis for developing organization's business blueprint as part of ERP system implementation project. Once these individual models for process integration, automation and optimisation are developed, they can be linked together for relevant applications.

Therefore, business processes designed for integration, automation and optimisation can provide improvements beyond BPR principles and can be extended with additional functionalities for detailed scheduling and process optimisation. Further, constraints of logical operators in process functions are removed for incorporating additional functionalities as part of process optimisation. For example, logical operator OR in the process integration can become an AND operator when process integration is enhanced further with process optimisation. In general, business process optimisation through additional functionalities primarily focus on three aspects: (i) finite loading of resources, (ii) forward planning at execution phase depending on requirements arising from uncertainty (iii) simultaneous planning of all involved.

3 Business Process Improvements Using Enhanced Process and Integrated Data Models

Business process improvements outlined in the proposed framework are sought through enhanced process and integrated data models. In this regard, the main focus is to enhance business process models across many functional applications and associate with integrated data models in ERP system environment. Thus, business processes are modeled using enhanced EPC methodology, incorporating additional components and relationships into functions of business processes.

Enhanced EPC methodology [10] integrates business process components with data elements and structures at the modeling level. It has been shown that it can allow further improvements of business processes using enhanced models for existing functions in business processes as well as providing the flexibility and maintainability of many process cycles. This enables improvements in processes for planning, control and execution of various components. Similarly, process automation can be enhanced

using unitary structures for functions associated with workflows in ERP system environment. The details of EPC methodology and process improvements with process automation are not presented here since it is beyond the scope of this paper. The next level of process improvements is to incorporate additional functionality as part of process optimisation, by removing some of the process constraints due to limitations within planning, control and execution cycles. One such limitation can be found in the production order cycle where there is no finite loading of resources at the time of planning of materials requirements.

Thus, production order cycle in ERP system environment is considered to demonstrate additional functionality for finite loading of resources, by eliminating manual capacity leveling of the current process, using enhanced EPC methodology. Further, production order cycle lacks simultaneous planning of all involved and forward planning of goods movements and other operations under uncertainty such as any breakdown of machines and unavailability of materials. Thus, various functions including production order creation and good movements can be considered as potential candidates for business process optimisation in ERP system environment. Thus, events related to those functions need to be handled using appropriate techniques at the time of order creation and linked with functions: finite loading in the case of overloaded capacity situation and forward planning of production, based on availability of materials leading to an event: materials and capacities availability, before releasing the production order. These additional events and functions can only be handled when the production order creation process is represented by enhanced EPC. Thus, enhanced EPC for production order cycle, incorporating additional functionality for finite loading of resources with relevant components based on unitary-structuring technique, improved process logics for compulsory events and associated functions and other possible functions is shown in Fig. 2.

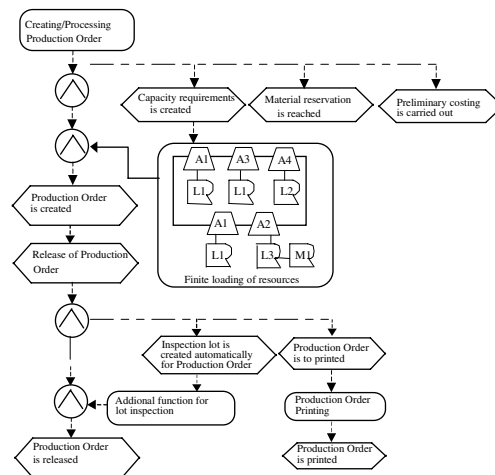


Fig. 2. Production Order Creation with Finite Loading Function

It can be noted from the production order creation process (Fig. 2) that the process is represented by a combination of existing functions and additional functions for optimizing the whole process. The enhanced process modeling and thereby process optimisation using additional functionality will result in additional transactions with workflows and optimisation aspects as necessary for some process cycles. For example, production order cycle demonstrated here, scheduling is based on availability of both materials and capacities at the time of order creation. Further, any unexpected situation during the execution phase can be handled since the production order cycle now contains not only hierarchical and sequential operations routings, but also other relationships between them. Overall, the production order cycle is improved through enhanced process models using improved integration, automation and optimisation.

The implementation requires a number of steps to be carried out within an ERP system environment. Further, it can result in improved transaction data where those functions are already built-in as part of the planning. First, business blueprint of ERP needs to incorporate enhanced process models, by expanding current functions and relationships beyond process elements. Once process map is implemented, business transactions are required to be modified, based on additional data and relationships. This requires some coding within the system, for combining current hierarchical data with enhanced process components. Additional functions involved in any process, as part of optimisation step, can be incorporated as a combination of additional functions within associated process areas. This step also requires some coding and is similar to that of incorporating expanded functions as part of process enhancements. Implementation of process improvements through a sample process cycle was carried out and will be reported later.

It can be noted from the production order cycle with additional functionality for finite loading (Fig. 2) that it is represented by various components and relationships at structural level. Since those components associated with functions are usually stored as master data in ERP systems and transaction data associated with those functions are based on master data, improving master and transaction data in ERP is vital for further improvements in processes in ERP. Improvements in master and transaction data are sought through structural integration, using unique method of unitary structuring technique [21].

4 Integrated Master and Transaction Data in ERP System Environment

In order to demonstrate the representation of production planning using unitary structure-based master and transaction data, a production planning business scenario involving various master data and transaction data in ERP system is considered. The business scenario is a make-to-stock (MTS) finished product with number of assemblies and raw materials, to be planned and executed using production management processes in an ERP system. Precisely, the finished product and assemblies are represented by a number of single-level BOMs and other required data including operations routings, work centres and cost centres attached to each work centre for costing purposes. Apart from master data described above, planning, control and execution generate and involve various transaction data including planned orders and production orders.

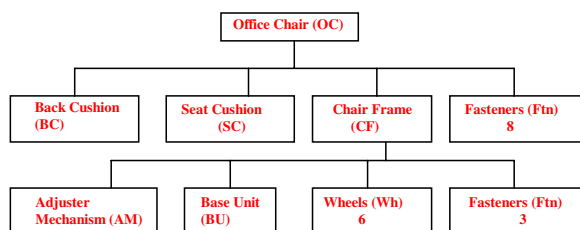


Fig. 3. Product Structure of Office Chair

In order to implement an MTS business scenario in an ERP system, it requires maintaining at least five key master data types: materials, bills of materials (BOMs), operations routings, work centers and cost centers. Thus, a product structure shown in Fig. 3 is considered as an example of MTS product for illustrating unitary structure-based master and transaction data. Since the product structure of office chair involves two single-level BOMs (one each for office chair and chair frame), master data for the office chair can be set in ERP systems using appropriate BOMs and operations routings. Since ERP system usually maintains single-level BOMs, there are two operations routings: one for each single-level BOM of the product structure. The details of resulting operations routings are shown in Tables 1 and 2.

Table 1. Operations Routing for Chair Frame

Opn. ID	Description	Work Centre	Set-up (Min)	Machine (Min/Unit)	Labour (Min/Unit)
10	Assemble AM (AAM)	R7	30	5	5
20	Painting AM (PAM)	R6	30	2	2
30	Cutting (Cu)	R10	30	2	2
40	Bending (Be)	R9	30	2	2
50	Welding (We)	R8	15	5	5
60	Painting (PBU)	R6	30	2	2
70	Assembly with BU (ABU)	R5	15	10	10
80	Inspection (ICF)	R4			5

Table 2. Operations Routing for Office Chair

Opn. ID	Description	Work Centre	Set-up (Min)	Machine (Min/Unit)	Labour (Min/Unit)
10	Fabric Cut (FBC)	R3	15	2	2
20	Ass. with BC (ABC)	R2	30	5	5
30	Fabric Cut (FSC)	R3	15	2	2
40	Ass. with SC (ASC)	R2	15	5	5
50	Final Ass. & Ins.	R1	15	10	5

When operations routings are maintained in ERP system, there are various functions associated with them including component allocation function for identifying the relationship between component and operation(s). The component allocation allows each material component assigned to an appropriate operation so that all the components are available for the relevant operation(s) at the time of execution of material and resource plans. In the case of the office chair (Fig. 3), base unit is required to be assigned to operation 30 (Cutting). Further, adjuster mechanism is assigned to operation 50 (Assemble AM) while wheels are assigned to operation 70 (Assemble AM/BU with CF). Similarly, all other components are required to be assigned to appropriate operation(s). It can be noted that all the master data with required functionality can be maintained for further processing during appropriate planning, control and execution processes. However, there are limitations within these data structures, which do not allow simultaneous and forward planning of all involved.

Apart from those two data structures, there are additional functions associated with the production order, including goods issues, goods receipts and order settlement. However, the production order as a transaction data at this level does not have the capability of adding those functions into a structured transaction data rather the information is copied into the order. Thus, the aspect of data being copied rather than being directly linked to the database, and need for separate functions during the order creation and beyond, limits the capabilities of smooth processing of production orders with these data elements. However, these issues can be handled using unitary structure-based transaction data.

Master data described above are candidates for integrated master data so that planning, control and execution can be streamlined and enhanced using additional functionality. For example, integrated operations routing with BOMs can eliminate component allocations in production order cycle. Further, longer lead times associated with sequential operations can be reduced when operations are integrated into BOMs using operations routing of both sequential and parallel operations. Thus, master data described above for this business scenario are integrated using unitary structures and presented next.

4.1 Master Data Integration Using Unitary Structures

Currently, the MTS scenario outlined above can be planned, controlled and executed using traditional ERP systems. However, there is lack of simultaneous planning, forward planning and finite capacity planning capabilities due to lack of true integration of data at the database level and inflexibility of planning techniques. As a first step of improvements for the planning, control and execution process, hierarchical BOM shown in Fig. 3 is integrated with sequential operations routing (Tables 1 and 2) to make a unitary structure-based one data structure. The resulting data structure is shown in Fig. 4.

It can be noted from Fig. 4 that original two routings and two BOMs are combined into one data structure. Further, it also provides built-in component allocation as part of the structure rather than separate activity in operations routing. In addition to integrated data structures, transaction data generated from this data can also be represented by unitary structure for effective execution of such transaction data. In many situations, transaction data are combined with various other events and associated

functions outside the functional application the original process belongs. Further, there are many activities and resources in such functions and events, which require synchronous planning of all involved. For example, production order creation process involves various other functions including goods movement at two levels: goods issues to the production and goods receipts from production. These functions/tasks are required to be carried out at the correct time for timely completion of the production order. Using unitary structure, these functions/tasks can be incorporated and planned for better outcome of the overall process rather than manual intervention required by current systems. Due to page limitation of this paper, the details of integrated transaction data using unitary structures are not presented here.

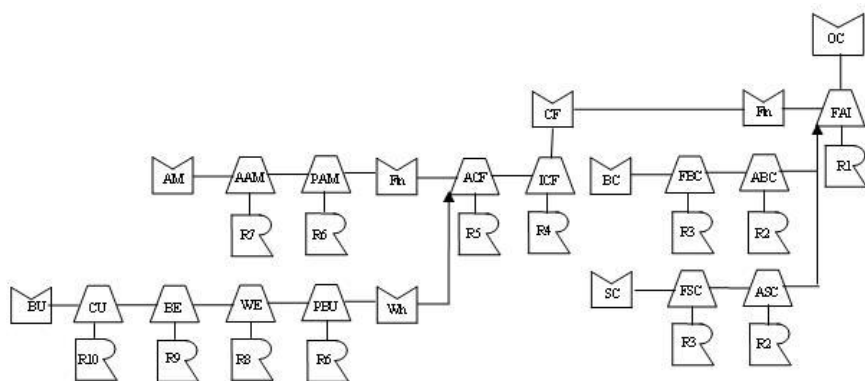


Fig. 4. Unitary Structure of BOM and Operations Routing for Office Chair

4.2 Production Planning and Execution Using Integrated Master and Transaction Data

This section illustrates planning and execution of various components using unitary structure-based master and transaction data of the business scenario described above. The planning of materials, activities and resources involved initiates with a set of planned independent requirements. For a given planned independent requirement, planning of materials, activities and resources are carried out using a scheduling method and scheduling paths developed earlier [22]. Scheduling paths determine the sequence for planning of components, taking all the relationships into consideration. In the case of backward scheduling, planning will start from the finished product while forward scheduling of all components starts from the last component in the lowest level of the BOM. Based on this approach, both bills of materials explosion and operations scheduling are carried out simultaneously. Thus, the complete planning and scheduling of components in the unitary structure shown in Figure 4 could result in start dates and times for activities and due dates and quantities for materials. The scheduling of operations routing for the same structure would result in operation start and finish times for all the operations, based on actual operation times rather than traditional lead times.

In this example, scheduling is based on the backward scheduling of all operations, with a due date and time of 28 November 2008 at 16.00 Hrs, for a quantity of 60 units. Operations scheduling is based on a working calendar of 8 hours, Monday to Friday between 08:00 hours and 16:00 hours. For the simplicity and numerical testing purpose, it is assumed that there is no break during the 8-hour shift. The scheduling of components of unitary structure-based master data is based on scheduling path depending on the type of scheduling (backward or forward) and the type of explosion depending on component relationships (parent-component, component-component and activity precedence) and component type (material, activity, resource, etc.). For backward scheduling, the scheduling path for each component in the unitary structure is a sequence of numbers starting from the office chair with a multiplication of 10. For the simplicity, it is assumed that each activity is associated with only one resource for both labour and machine categories. Thus, scheduling sequence is simplified by one sequence number for each activity and resource. All the components are planned in terms of start/finish dates and times for activities, due dates/times for resources, using BOM explosion and operations scheduling. The resulting scheduling path (identified by sequence numbers) for the entire unitary structure (Fig. 4) and planning results are shown in Table 3.

Table 3. Exploded Quantity and Time Schedule for each Component (Backward Schedule)

Seq. Number	Component	Exploded		Ass. Qty	Due Date & Time
	Name	Qty	Duration (HRS:MIN)		
10	Office Chair (OC)	60		60	28 Nov. 08 16:00
20	Final Adj. & Ins. (FAI)		15:15		28 Nov. 08 16:00
30	Fabric & SC ass. (ASC)		10:15		27 Nov. 08 08:45
40	Fabric cut for SC (FSC)		4:15		26 Nov. 08 14:30
50	Seat Cushion (SC)	60			26 Nov. 08 10:15
60	Fabric & BC ass. (ABC)		10:30		27 Nov. 08 08:45
70	Fabric cut for BC (FBC)		4:15		26 Nov. 08 14:30
80	Back Cushion (BC)	60			26 Nov. 08 10:15
90	Fastener (Ftn)	480			27 Nov. 08 08:45
100	Chair Frame (CF)	60		60	27 Nov. 08 08:45
110	Chair Frame Ins. (ICF)		5:00		27 Nov. 08 08:45
120	Ass. AM/BU in CF (ACF)		20:15		26 Nov. 08 11:45
130	Wheel (Wh)	360			23 Nov. 08 15:45
140	Painting BU (PBU)		4:30		23 Nov. 08 15:45
150	Welding BU (We)		10:15		23 Nov. 08 11:15
160	Bending BU (Be)		4:30		22 Nov. 08 09:00
170	Cutting BU (Cu)		4:30		21 Nov. 08 12:30
180	Base Unit (BU)	60			21 Nov. 08 08:30
190	Fastener (Ftn)	240			23 Nov. 08 15:45
200	Painting AM (PAM)		4:30		23 Nov. 08 15:45
210	Assemble AM (AAM)		10:30		23 Nov. 08 11:15
220	Adj. Mechanism (AM)	60			22 Nov. 08 08:45

In this example, the scheduling of components starts from the office chair (seq. no. 10) for a requirement of 60 units. Scheduling of components would result in exploded quantities for both materials and activities. The exploded material quantity is based on BOM explosion while exploded activity duration is a combination of setup, labour and machine times. Since there is only one resource unit per each activity, total resource requirements are same as those of activity duration. The exploded activity duration can be calculated by:

$$\text{Total Activity Duration} = \left\{ \text{Setup} + \left(\frac{\text{Assembly Qty}}{\text{Base Qty}} \right) * ((\text{Machine Time / Unit}) + (\text{Labour Time / Unit})) \right\} \quad (1)$$

For example, activity duration for FAI and its associated resource is 15 Hours and 15 Min, which is the sum of setup time of 15 Min and 10 hours of machine (60*10 Min) and 5 hours of labour (60*5 Min).

It could be noted from planning results (Table 3) that many types of components involved in a unitary structure in manufacturing environment could be planned using two approaches as discussed above. It can also be noted that raw material of Base Unit (BU) is required around one week earlier (21 Nov. 2008 08:30) than the scheduled due date of completion of the office chair, making total lead-time around one week. Similar to scheduling of unitary structure-based components, production activity control (execution of material and capacity plans) can be carried out using unitary structure-based transaction data. In this case, the primary transaction data for execution is the production order. The resulting production order for the office chair is shown in Figure 4 as described earlier. In this case, production order is based on the unitary structure-based master data and consists of additional activities and resources for good movements and component allocation. Thus, execution of production order is based on due date and quantity derived from the scheduling of components and can provide scheduling of goods movement activities and resources using scheduling paths and explosion methods as described earlier. The complete production order execution including scheduling of goods movement activities and resources are not presented here due to page limitation of this chapter.

The methodology using integrated master and transaction data, when implemented in an ERP system, enables planning of many components and could result in planned orders and/or purchased requisitions depending on the type of components. Further, production order cycle can be improved through eliminating manual planning of goods movement activities and resources. Implementation of the methodology is being carried out and results would be published in the future. Thus, the integration of data structures for both master and transaction data provide flexibility of dynamically changing situation within many planning, control and execution cycles and allows critical path to be dynamically decided based not only on activities but also on availability of materials and resources during the execution phase. The methodology for planning these structures is not discussed here since it is beyond the scope of this paper.

5 Conclusions

The paper presented a framework for business process improvements using enhanced process and integrated data models for process integration, automation and optimisation.

Business processes are based on enhanced EPC methodology, incorporating many components, relationships and links and additional functionality for process optimisation. Main features of the framework are (i) enhanced process models using enhanced EPC methodology and (ii) integrated data models using various data elements, structures and relationships. The proposed framework is illustrated using a business process and a numerical example within production planning functional area of ERP system environment. It is shown that process improvements can be achieved through enhanced process models and integrated data models. The enhanced process and integrated data models, when implemented in an ERP system, can provide streamlined transactions combining process integration, automation and optimisation. In addition, it is capable of providing visibility, flexibility and maintainability for further improvements, in particular in process optimisation using additional functionality with enhanced process and integrated data models.

References

1. Beretta, S.: Unleashing the Integration Potential of ERP Systems: The Role of Process-Based Performance Measurement Systems. *Business Process Management Journal* 8(3), 254–277 (2002)
2. Sandoe, K., Corbitt, G., Boykin, R.: *Enterprise Integration*. John Wiley & Sons, New York (2001)
3. Sumner, M.: *Enterprise Resource Planning*. Pearson Prentice Hall, New Jersey (2005)
4. Hammer, M.: Re-engineering work: don't automate – obliterate, pp. 104–112. *Harvard Business Review* (July–August 1990)
5. Peppard, J., Rowland, P.: *The Essence of Business Process Re-engineering*. Prentice Hall, London (1995)
6. Koch, C.: BPR and ERP: realizing a vision of process with IT. *Business Process Management Journal* 7(3), 258–265 (2001)
7. Bhatt, G.D.: An Empirical Examination of the Efforts of Information Systems Integration on Business Process Improvement. *International Journal of Operations & Production Management* 20(11), 1331–1359 (2000)
8. Themistocleous, M., Irani, Z., O'Keefe, R.M.: ERP and application integration. *Business Process Management Journal* 7(3), 195–204 (2001)
9. Barber, K.D., et al.: Business-process modeling and simulation for manufacturing management – A practical way forward. *Business Process Management Journal* 9(4), 527–542 (2003)
10. Samaranayake, P.: Business Process Integration with Data Structures in Enterprise Resource Planning Environment. In: Parkinson, P.S., Shutt, P.J. (eds.) *British Academy of Management Annual Conference*, pp. 1–9. Niche Publications UK Ltd., Harrogate (2003)
11. Persona, A., Regattieri, A., Romano, P.: An integrated reference model for production planning and control in SMEs. *Journal of Manufacturing Technology Management* 15(7), 626–640 (2004)
12. Sandhu, M.A., Gunasekaran, A.: Business process development in project-based industry – A case study. *Business Process Management Journal* 10(6), 673–690 (2004)
13. Metaxiotis, K., et al.: Goal directed management methodology for the support of ERP implementation and optimal adaptation procedure. *Information Management and Computer Security* 13(1), 55–71 (2005)

14. Martin, I., Cheung, Y.: Business process re-engineering pays after enterprise resource planning. *Business Process Management Journal* 11(2), 185–197 (2005)
15. Serrano, A., Hengst, M.D.: Modelling the integration of BP and IT using business process simulation. *Journal of Enterprise Information Management* 18(6), 740–759 (2005)
16. Samaranayake, P., Chan, F.T.S.: Business Process Integration and Automation in ERP System Environment – Integration of Applications and Workflows. In: *The 2nd International Conference on Information Management and Business*, Sydney, Australia (2006)
17. Clegg, B.: Business process oriented holonic (ProOH) modeling. *Business Process Management Journal* 12(4), 410–432 (2006)
18. Kuechler Jr., W., Vaishnavi, V.K., JKuechler, D.: Supporting optimization of business-to-business e-commerce relationships. *SDecision Support Systems* 31, 363–377 (2001)
19. Vergidis, K., et al.: Optimisation of business process designs: An algorithmic approach with multiple objectives. *International Journal of Production Economics* 109, 105–121 (2007)
20. Worley, J.H., et al.: Implementation and optimization of ERP systems: A better integration of processes, roles, knowledge and user competencies. *Computers in Industry* 56, 620–638 (2005)
21. Woxvold, E.R.A.: Extending MRP II Hierarchical Structures to PERT/CPM networks. In: *35th International Conference - American Production and Inventory Control Society*, Canada (1992)
22. Samaranayake, P.: Scheduling Paths for Merged CPM Networks with MRP Bills-of-Materials. In: *Toncich, D. (ed.) Profiles in Industrial Research*, pp. 1–9. Industrial Research Institute Swinburne (1998)

Computing the Cost of Business Processes

Partha Sampath and Martin Wirsing

Institute of Computer Science,
Ludwig-Maximilians-University, Oettingenstr. 67, 80538 Munich, Germany
`partha.sampath@capgemini-sdm.com, wirsing@lmu.de`
<http://www.pst.ifi.lmu.de/>

Abstract. Computing the cost of a Business Process is a complicated and cumbersome process. Magnani and Montesi have proposed a concept of evaluating the cost by providing extensions to the Business Process Modelling Notation (BPMN). In this paper we propose a method by which the cost of a Business Process is calculated by considering the cost and reliability of each action or task in the process. This method breaks the Business Process, represented using BPMN, into repetitive patterns and a cost and reliability factor for each of these patterns is calculated. The method calculates the overall cost, reliability and the cost incurred to achieve one successful execution of the Business Process; the Business Cost of the process.

Keywords: BPMN, BPD, Cost, Reliability.

1 Introduction

BPMN, a standardized graphical notation for drawing Business Process Diagrams (BPD) in a workflow [2], concentrates mainly on representation and does not deal with the quantitative aspects such as cost and reliability of a Business Processes. The amount of money spent so as to execute a Business Process once is the cost of the process. Because a Business Process achieves its cost doesn't mean that the business value of the process has been achieved. To achieve business value we try to achieve higher reliability. Reliability can be interpreted from a technical perspective e.g. network availability and from a business perspective e.g. rate at which the business goal is achieved. There have been models by which the costs can be calculated from BPMN but they do not take reliability into account, especially from the business perspective. Magnani and Montesi [1] in their study have proposed a method for representing and calculating the cost of BPD's. We base ourselves on this study.

2 Representation of Cost and Reliability in BPD's

To calculate the Business cost of a process we only consider elements which fall under Flow Objects such as Activities and Gateways as these represent some action or job that needs to be done. The rest such as Swimlanes etc do not play

a role in cost calculation. To each artefact, with a simple extension in the form of textual property, we assign cost and reliability as properties. We define these as:

$$Cost = C, \text{ where } C \geq 0 \quad (1)$$

$$Reliability = R, \text{ where } 0 < R \leq 1. \quad (2)$$

Constructs such as Gateways have cost = 0 and the reliability = 1. In the rest of this paper we elaborate on a method by which we calculate the Business Cost of a task and the whole BPD by patterns.

3 Calculation Method

We consider different patterns of Flow Objects and calculation of costs attached to them. We start with the most elementary combinations and move down to the patterns which have higher complexity. As a basis we consider two elementary scenarios:

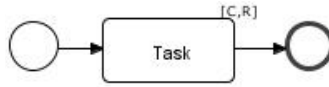


Fig. 1. BPD with one task

Scenario 1: BPD with one single task with no compensation having cost C and reliability R as shown in the Fig. 1. The Business Cost in this case is the result of dividing the cost by the reliability.

$$BusinessCost = C/R \quad (3)$$

We make the assumption that we always pay for a service to use it immaterial of its success or failure. Hence, the Business Cost is always going to be higher than the cost when the reliability of the task is less than 1.

$$BusinessCost = Cost, Reliability = 1 \quad (4)$$

$$BusinessCost > Cost, 0 < Reliability < 1 \quad (5)$$

Scenario 2: BPD with one single task having a cost C and reliability R and with an error flow as shown in the Fig. 2.

The error flow is invoked when the task fails and hence we consider the task together with its error flow. As a result the cost of the task will be the same as the Business Cost of this task which can be represented by the equation:

$$BusinessCost = C + ((1 - R) * (C + C(E)))/R \quad (6)$$

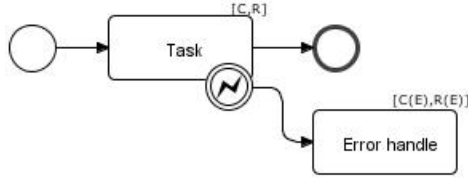


Fig. 2. BPD with one task and an error flow

4 Patterns for Cost Calculation

When we have BPD's with a number of tasks invoked in different manners, the cost, reliability and the Business Cost can be calculated by recognizing patterns which are repetitive. The values generated for each of these patterns put together gives the Business Cost of the overall Business Process. In this section we look at the four common patterns which we come across in Business Processes and derive the cost, reliability and Business Cost for each one.

4.1 Pattern 1: n Tasks in a Sequential Order

We are considering a pattern/process with n tasks (with no compensation) in a sequential order as shown in Fig. 3. The cost of the BPD is the summation of the costs and the reliability of the BPD is the product of the reliabilities of all the events. Equation (3) gives us the Business Cost for a single task. Business Processes contains tasks which are mutually inclusive i.e. the execution of a task is dependent on the successful completion of the tasks before it. Equation(3) can be used for patterns containing tasks which are mutually exclusive in nature. In case of a pattern where we have n tasks as the one shown in Fig. 3, the Business Cost is calculated by a recursive way, depending upon the number of tasks in the pattern. We consider n tasks where each task comes with a cost and reliability.

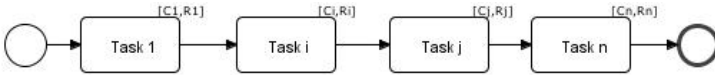


Fig. 3. n tasks in a sequential order

In case of one task in the pattern the Business Cost would be

$$BusinessCost(1, 1) = C1/R1 \quad (7)$$

In case of two tasks

$$BusinessCost(1, 2) = C2/R2 + (C1/R1)/R2 = (C2 + (C1/R1))/R2 \quad (8)$$

$$BusinessCost(1, 2) = (C2 + BusinessCost(1, 1))/R2 \quad (9)$$

Hence, in case of a sequential/serial pattern with n tasks:

$$BusinessCost(1, n) = (Cn + BusinessCost(1, n - 1)) / Rn \quad (10)$$

4.2 Pattern 2: n Tasks in a Parallel Order

We are considering a pattern/process with n tasks (with no compensation) in a parallel order as shown in Fig. 4. In a BPD with parallel tasks, each flow is a

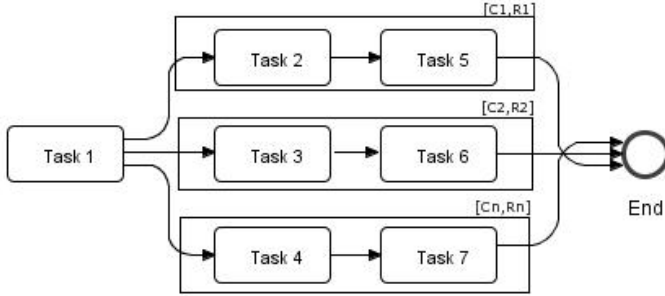


Fig. 4. n tasks in parallel order

sequential flow with one or more tasks. For each of the sequential patterns the cost, reliability and the Business Cost is calculated as shown in Pattern 1. The resulting cost and reliability of this parallel pattern then would be:

$$Cost = \Sigma Ci, \text{ (} Ci \text{ is the cost of each flow in the parallel flow)} \quad (11)$$

$$Reliability = Minimum(R) \quad (12)$$

$$BusinessCost = \Sigma BusinessCost(i) \text{ (} i \text{ is the pattern)} \quad (13)$$

4.3 Pattern 3: Conditional Branching

We consider a pattern/process with a conditional branching leading to different execution paths. This is same as in Pattern 1 with sequential tasks. Nevertheless a probability has to be attached to each flow out of the Gateway. The corresponding cost of the path is then multiplied by the probability which will lead to the cost of the whole branching.

$$Cost = \sum Pi * Cost(i), \text{ (} Pi \text{ is the probability of taking path } i) \quad (14)$$

$$Reliability = \sum Pi Ri, \text{ (} Ri \text{ is the reliability of path } i) \quad (15)$$

$$BusinessCost = \sum Pi * BusinessCost(i) \quad (16)$$

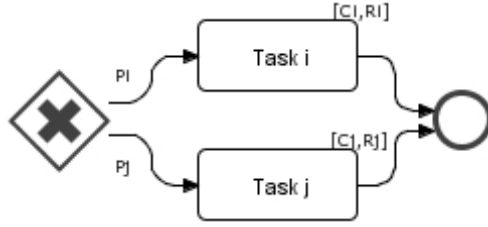


Fig. 5. BPD with conditional branching

4.4 Pattern 4: "n" Successive Possibilities

We are considering a pattern/process with n different services each performing the same function. Let us assume a BPD which tries to book a room in n hotels such that $\text{Cost}(\text{Hotel } i) = C_i$ and $\text{Reliability}(\text{Hotel } i) = R_i$. The token first talks to the first hotel and then to the second and so on. The cost, Business Cost and reliability depend on the number of hotels and the order in which they are talked to. In this scenario we see that a hotel, Hotel i , is contacted when all $i - 1$ hotels before it have already been contacted. Hence the cost of talking to Hotel i is not just its cost but the accumulation of costs of the first $i - 1$ hotels. We call this cost the Actual Cost. In Table 1 we calculate the cost, Actual Cost, Business Cost and reliability in case of 1 hotel, 2 hotels and n hotels. The results can be represented with following equations:

Table 1. Actual and Business Cost

No. Hotels	Reliability	ActualCost (AC)	BusinessCost
Hotel(1,1)	$1 - (1 - R_1)$	C_1	C_1/R_1
Hotel(1,2)	$1 - (1 - R_1) * (1 - R_2)$	$AC(\text{Hotel}(1,1)) + C_2(1 - R_1)$	$AC(1,2)/Rel_2$
Hotel(1,n)	$1 - ((1 - R_1) * (1 - R_2) .. (1 - R_n))$	$AC(\text{Hotel}(1,n-1)) + C_n(1 - R(n-1))$	$AC(1,n)/Rel_n$

$$\text{ActualCost}(N_i, N_n) = \text{ActualCost}(\text{Hotel}(1, n - 1)) + C_n(1 - R(n - 1)) \quad (17)$$

$$\text{BusinessCost}(N_i, N_n) = \text{ActualCost}/\text{Reliability} \quad (18)$$

$$\text{Reliability}(R_1, R_n) = 1 - ((1 - R_1)(1 - R_2) .. (1 - R_n)) \quad (19)$$

5 Example: Breaking a BPD into Patterns

We consider a Business Process for booking a hotel, flight and taxi for a customer as an example as shown in Fig. 6. The tasks Authenticate customer and Validate input are tasks in sequential order and hence are in Pattern 1. The tasks Book Hotel, Book Flight and Book Taxi are in parallel and hence are in Pattern 2. The tasks Send confirmation and Send rejection are controlled through a Gateway

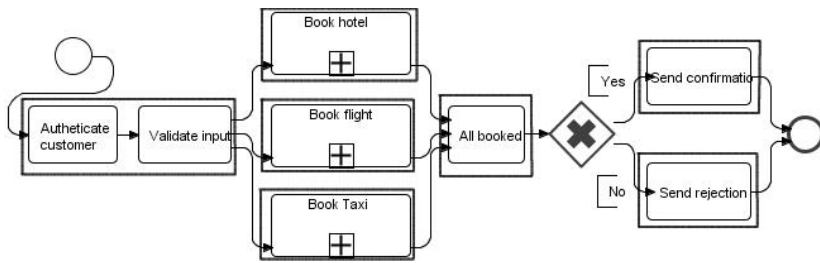


Fig. 6. Sample BPD

and are hence in Pattern 3. In the task Book Hotel, there could be n possible hotels with which a reservation can be done, hence this falls into Pattern 4. The Business Cost and cost of the BPD is the summation of the Business Cost and cost of the patterns whereas the reliability is the product of the patterns.

6 Further Work and Conclusion

The concept of compensation in BPMN doesn't match with the theoretical foundations for compensation such as Sagas [3] which deals with transaction compensations and hence has not been considered in this study. Calculating the impact of the cost of compensation on the Actual Cost or the Business Cost of the BPD deviates from the method represented here as it is dependent on the task that throws the trigger and its reliability. In this paper, we have presented a method by which the cost, Business Cost and reliability of a Business Process can be calculated considering the reliability of each of the artefacts. We are interested in having a standardised and parameterized approach towards cost calculation of BPD which would play a decision making role in designing and developing Business Processes.

References

1. Magnani, M., Montesi, D.: Computing the Cost of BPMN diagrams. Technical Report UBLCS-07-17 (2007)
2. OMG. Business process modeling notation specification (2006)
3. Bruni, R., Melgratti, H., Montanari, U.: Theoretical Foundations for Compensations in Flow Composition Languages. In: 32nd ACM SIGPLAN-SIGACT, pp. 209–220 (2005)
4. Clark, A., Gilmore, S.: Evaluating Quality of Service for Service Level Agreements. LNCS, pp. 181–194. Springer, Heidelberg (2007)
5. Vanhatalo, J., Völzer, H., Koehler, J.: The Refined Process Structure Tree. In: Dumas, M., Reichert, M., Shan, M.-C. (eds.) BPM 2008. LNCS, vol. 5240, pp. 100–115. Springer, Heidelberg (2008)

Formalizing Computer Forensics Process with UML

Chun Ruan and Ewa Huebner

School of Computing and Mathematics
University of Western Sydney, Penrith South DC, NSW 1797 Australia
{c.ruan,e.huebner}@uws.edu.au

Abstract. This paper introduces modeling methodologies to computer forensics to provide formalism and structured approach to computer forensics activities. It studies how to use UML diagrams to model and visualize various aspects of a computer forensics system. It first applies UML to model the basic components of a computer forensic process and their relationships. It then uses UML to further visualize the activities carried out for each component. The formal graphical model provides a well-defined and straightforward semantics to the computer forensics process making it easier to understand by various parties involved.

Keywords: Computer forensics process, system modeling, UML.

1 Introduction

Computer forensics emerged in response to the increasing involvement of computer systems in crimes, as an object of crime or as a tool of crime. Computer forensics has been defined as the use of scientifically derived and proven methods towards the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations [3]. Computer forensic techniques are used for conducting examination of computer systems to help find out what had happened, when and where it happened, how and why it happened, and who was involved.

One important issue in computer forensics is the process or methodology adopted by people in conducting an investigation. Overlooking one step or interchanging some steps may lead to wrong interpretations and conclusions. Various process models have been proposed over the years for computer forensics [1]. Pollitt identified four stages for a computer forensics process: acquisition, identification, evaluation and admission as evidence [2]. In 2001, seven core steps of this process were identified: identification, preservation, collection, examination, analysis, presentation and decision [3]. Reith et al proposed a nine step model: identification, preparation, approach strategy, preservation, collection, examination, analysis, presentation and returning evidence [4]. Carrier et al defined 17 phases classified into five groups: readiness, deployment, physical crime scene investigation, digital crime scene investigation and review phases [5]. Beebe et al proposed a multi-tier framework to guide the computer forensics investigation. The first-tier phases include preparation, incident response,

data collection, data analysis, incident closure and findings presentation. Each phase contains several sub-phases to include various types of tasks corresponding to various types of crime [6]. Leong identified eight different roles and their responsibilities in a computer forensics investigation [7]. Bogen et al proposed a multi-view framework and used UML modeling to describe them [8]. Carroll et al listed six key elements of computer forensics and used flowcharts to describe the computer forensic analysis methodology [9].

Most proposed models are presented in text, with some assisted with tables. There is a good effort in [8] where UML modeling is introduced. Flowcharting was used recently in [9] to describe the activities, but many aspects of the system, such as the system's overview of the process, are difficult to describe. This lack of formalization could result in an ambiguous semantics of the model if sufficient level of detail is not captured. Alternately the model could become too complicated to understand due to the lengthy description of details. What is needed is a formal model to specify the methodology used in the computer forensics analysis, which can provide a clear and well-defined semantics. A graph based model has the advantage of visualizing the process requirements, thus making the model more understandable to various parties involved. In this paper, we propose a UML model to formally specify the general computer forensics process. The UML [10] is a de-facto standard modeling language for analysis and design of software systems issued by the Object Management Group (OMG). The primary purpose of the UML is visualizing. In this paper, we use the following UML diagrams: use case diagram, interaction overview diagram, class diagram and activity diagram.

2 Necessary Parts for the Computer Forensics Process

In a computer forensics investigation process, the people involved (actors) usually come from three areas: domain area, computer forensics area, and legal area. The participants in investigation can be further categorized into individual roles [7]: case leader, is the organizer of the entire process. In the domain area, the system owner is usually the victim and sponsor of the case. The system administrators are also involved to provide the necessary information needed by the forensic investigators. In the forensics area, the roles involved are the digital forensics specialist, responsible for defining the investigation strategy, the digital forensics investigator, responsible for the investigation, and the digital forensics analyst. In the legal area, the people involved include the legal advisor who provides the initial legal support to the investigation team, and the legal prosecutor. Finally, the audience may come from the corporate structure, if the investigation is conducted for internal disciplinary hearings, or the jury and judge if the case proceeds to a legal dispute.

In terms of the computer forensics investigation process, the following basic components are well accepted in the current literature, regardless of the situation:

- **Preparation:** prepare tools, techniques, search warrants and other support.
- **Collection:** identify, record, and acquire relevant data from the possible sources. The data collection should follow the accepted guidelines and procedures that preserve the integrity of the data.

- **Examination:** perform in-depth systematic search for the evidence; forensically process large amounts of collected data to assess and extract data of particular interest, while preserving the integrity of the data.
- **Analysis:** analyze the results of the examination, using forensically sound methods and techniques, and draw conclusions based on the evidence found. Examples of analysis types include [5] Media, Media Management, File System, Network, Memory, and Application, for example. operating system files, executable images, graphic images, and video files.
- **Reporting:** present the results of the analysis, summarize and explain conclusions. This may also include explaining how tools and procedures were selected, describing actions performed etc.

3 UML Model for Forensics Process

In this section, we will show how to use UML diagrams to model and visualize the requirements stated in the previous section.

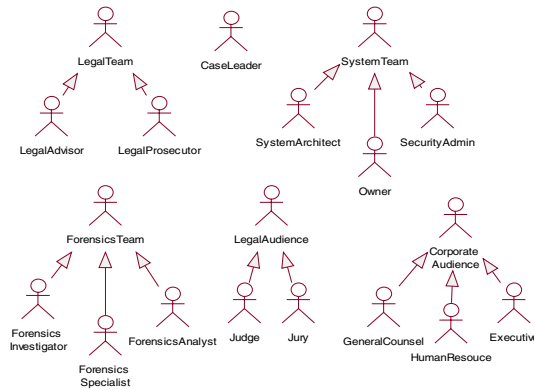


Fig. 1. Actors and their hierarchies in the computer forensics process

Actors: Firstly, we use Figure 1 to describe the Actor hierarchy, which represents the roles involved in the computer forensics investigation. The hierarchy reflects the inheritance relationship specified by arrows. For example, a forensics analyst inherits all the characteristics of a forensics team which again inherits all the characteristics of a Role. Inheritance can greatly reduce the complexity of the computer forensics process specification. Here forensics team, legal team, system team, legal audience and corporate audience are all abstract actors.

Use case diagram: In Figure 2, the first use case diagram gives an overview of the forensics process in the legal setting. Each use case represents one basic phase in the process, and the actors represent various roles involved. The association between the actor and the use case further shows who is involved in each phase. The associated use case documentation in UML can describe the detailed activities conducted in each process phase.

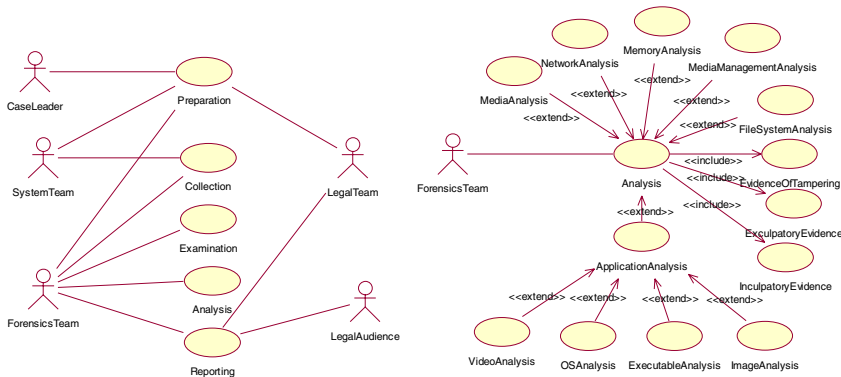


Fig. 2. Use case diagrams for computer forensics process and for forensic analysis

The other use case diagram in Fig. 2 presents the high level view of the forensic analysis. The main actor is the forensics team who is associated with the analysis use case. The diagram shows a number of extensions for the use case analysis. For any analysis, three types of evidence are looked for: inculpatory evidence that supports a given theory, exculpatory evidence that contradicts a given theory, and evidence of tampering that is not specific to a given theory. Each type of function is represented by a use case and has an include relationship with the use case analysis.

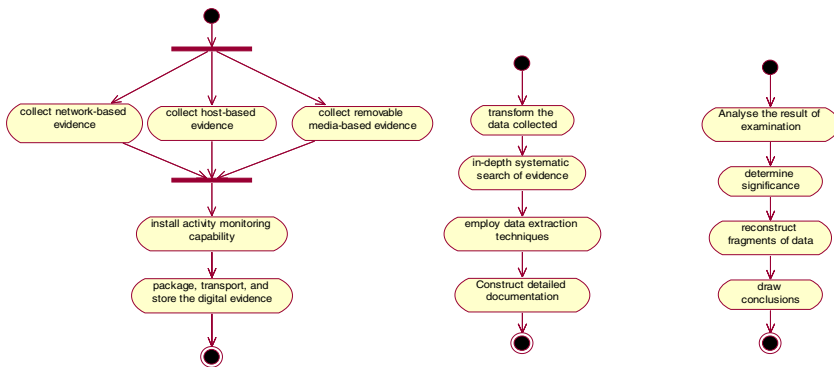


Fig. 3. Activity diagrams for collection, examination and analysis

Activity diagram: We further use three activity diagrams in Fig. 3 to visualize the activities for the three use cases: collection, examination, and analysis, simplified due to the space limit. In reality, they could be much more complicated.

Interaction overview diagram: For the sequence relationships between the 5 basic phases, in a computer forensics process, we use the interaction overview diagram (Fig. 4) to represent the high level flow. Note that the two phases of examination and analysis can be repeated several times before a conclusion is reached.

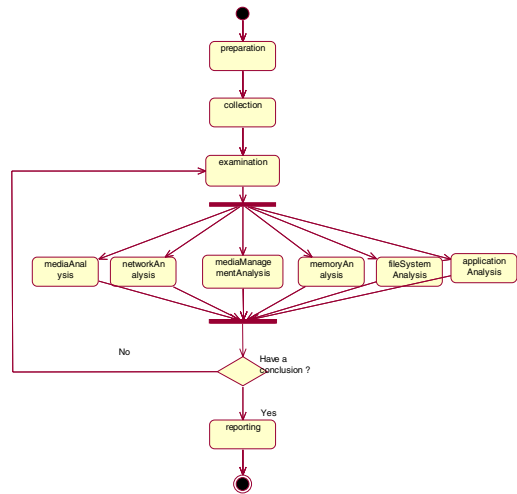


Fig. 4. The interaction overview diagram for the computer forensics process

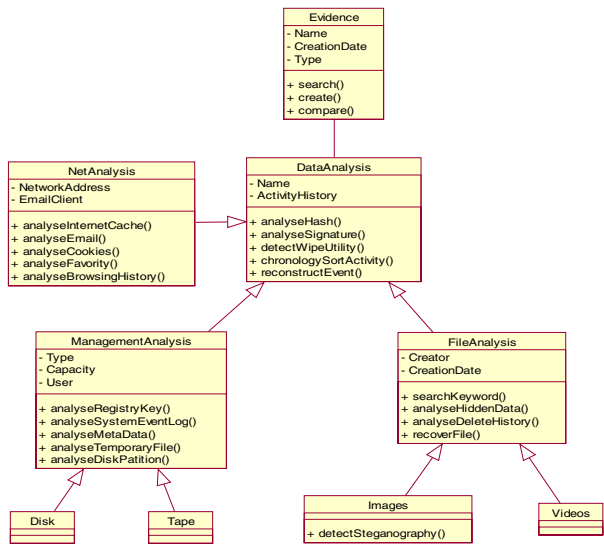


Fig. 5. Class diagram for data analysis

Class diagram: Fig. 5 shows how to use the class diagram to represent the structural information about the classes and their relationships. The inheritance relationship helps to reduce the redundancy, and therefore reduces the complexity of the process.

4 Conclusion

In this paper, we have proposed a UML based model to formalize the computer forensics process. The UML style diagrams have been used to describe various aspects of the process, from process overview to detailed phase activities, from functions flow to object state representation. The graphical model provides the advantages of well-defined semantics, and straightforward denotations which are understandable by various parties involved. The strong modeling mechanism and flexible model development tools ensure completeness of the process, and allow for adjusting the level of detail according to current needs.

For the future work, we will investigate and formalize the complete incident response management process, which includes the computer forensics process already formalized in this paper. This is of crucial importance to all organizations using ICT as a base for their operations. No such formal model currently exists, with the only support being a set of guidelines issued by the ISO (International Organisation for Standardisation) in 2004 [11].

References

1. Pollitt, M.M.: An Ad Hoc Review of Digital Forensic Models. In: The Second international Workshop on Systematic Approaches to Digital Forensic Engineering, pp. 43–54. IEEE Computer Society, Washington (2007)
2. Pollitt, M.: Computer Forensics: an Approach to Evidence in Cyberspace. In: Proceedings of the National Information Systems Security Conference, Baltimore, MD, pp. 487–491 (1995)
3. Palmer, G.: A Road Map for Digital Forensics Research, Utica, NY, Technical report DTR-T001-0 (2001)
4. Reith, M., Carr, C., Gunsch, G.: An Examination of Digital Forensic Models. *IJDE* 1(3), 1–12 (2002)
5. Carrier, B., Spafford, E.: Getting Physical with the Digital Investigation Process. *International Journal of Digital Evidence* 2(2) (2003)
6. Beebe, N., Clark, J.: A Hierarchical, Objectives-Based Framework for the Digital Investigations Process. *Digital Investigation* 2(2), 146–166 (2005)
7. Jeong, R.S.C.: FORZA - Digital Forensics Investigation Framework that Incorporate Legal Issues. *Digital Investigation* 3(suppl.1), 29–36 (2006)
8. Bogen, A.C., Dampier, D.A.: Unifying Computer Forensics Modeling Approaches: a Software Engineering Perspective. In: The First International Workshop on Systematic Approaches to Digital Forensic Engineering (2005)
9. Carroll, O.L., Brannon, S.K., Song, T.: Computer forensics: digital forensic analysis methodology. *Computer Forensics* 56(1), 1–8 (2008)
10. Unhelkar, B.: Practical Object Oriented Analysis. Thomson (2003)
11. ISO/IEC TR 18044, Information security incident management, ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) (2004)

Improving Product Usage Monitoring and Analysis with Semantic Concepts

Mathias Funk¹, Anne Rozinat², Ana Karla Alves de Medeiros², Piet van der Putten¹,
Henk Corporaal¹, and Wil van der Aalst²

¹ Dept. of Electrical Engineering, Eindhoven University of Technology, The Netherlands
{m.funk,p.h.a.v.d.putten,h.corporaal}@tue.nl

² Information Systems Group, Eindhoven University of Technology, The Netherlands
{a.rozinat,a.k.medeiros,w.m.p.v.d.aalst}@tue.nl

Abstract. Nowadays, complex electronic products, such as DVD players or mobile phones, offer a huge number of functions. As a consequence of the complexity of the devices, customers often have problems to use such products effectively. For example, it has been observed that an increasing number of technically sound products is returned due to, e.g., interaction problems. One possible root cause of this problem is that most product development processes are still too technology-driven, i.e., potential users are brought into contact with the product only at a very late stage. If early consumer tests are carried out, then these typically aim at abstract market evaluations rather than formulating concrete requirements towards the functionality of the product. As a result, products often have little meaning or relevance to the customers. Therefore, we need better ways to involve users in the development of such products. This can be achieved by observing product usage in the field and incorporating the gained knowledge in the product creation process. This paper proposes *an approach to build automatic observation modules into products, collect usage data, and analyze these data by means of process mining techniques exploiting a novel semantic link between observation and analysis*. This link yields two main benefits: (i) it adds focus to the potential mass of captured data items; and (ii) it reduces the need for extensive post-processing of the collected data. Together with the framework's flexibility to change observation modules remotely on-the-fly, these benefits speed up the information feedback cycle towards development.

Keywords: Product monitoring, log analysis, process mining, ontologies, semantic process mining.

1 Introduction

Complex electronic products, both for private consumers and professional users, are hard to specify and design as no real information is available about the potential customers' expectations and needs. Meeting these expectations is, however, crucial as nowadays customers can choose among a wide variety of products, and will more easily reject products that do not suit their needs. A symptom of this problem is, for example, that an increasing number of technically sound products is being returned [1]. At the same time, it is not possible to perform lengthy user studies as there is a strong pressure

on ‘time to market’. Moreover, it is difficult to gradually improve products by incorporating user feedback from the field as often only very few generations of the same product are made (to be replaced by new, more innovative products). In short, customers are becoming more demanding, whereas product development must be done with fewer iterations.

One way to ensure that products will suit the needs of potential customers is to involve these people as early as possible in the development process. This can be achieved by letting potential users test early prototypes, and to incrementally incorporate the gained knowledge into the product under development. However, to make this approach applicable in practice, two conditions need to be fulfilled.

1. It needs to be *feasible* to perform the tests in the first place, i.e., it should fit into today’s challenging development cycles.
2. The collected test data needs to be *useful*, i.e., valid (“Does this reflect our potential customers?”) and relevant (“Is this what we want to know?”).

To address the first condition, the test data needs to be collected and fed back to the development team as fast and automatically as possible. As we will demonstrate later, *our approach is supported by a tool chain that allows for seamless data collection, processing and analysis with a high degree of automation*. Addressing the second condition is more difficult as data quality depends on a variety of parameters. For example, to obtain valid data one needs to choose test users that reflect the actual target group. However, one common problem is that early user tests are often performed in non-representative environments, and that people do not behave normally as they feel observed. *Our approach allows for data collection from testers using the product in their habitual environment*. For example, test products are given to users who unpack, install and use the devices at home. The products themselves record usage information and automatically deliver it to the respective development unit in the company. This way, tests can easily run several weeks, and thus cover different phases of use [2]. Research has shown that the long-term usage behavior is often quite different from the behavior during the first few hours after unpacking the product. Finally, to ensure that the right data is collected, *we allow the observation logic to be changed dynamically by the development team, i.e., while the test is running*. This way, truly iterative data collection and analysis becomes possible. Furthermore, a visual approach to specifying the observation logic is taken to make it accessible to the (mostly non-technical) people that have an interest in the data collection process. These are, for example, product managers, quality engineers, interaction designers, or user interface developers.

With the aim to further increase both the feasibility and the usefulness of product usage observation, we extend the above-described approach by an important aspect: in this paper, *we establish a semantic link between the observation and analysis phase*. More precisely, we allow to semantically annotate the logged data during the specification of the observation logic, and these semantic annotations are preserved and actively leveraged in the analysis phase (by *semantic process mining techniques*). So-called *ontologies* [3], which are representations of a set of concepts within a domain and the relationships between those concepts, are used to define these semantic aspects. To allow different views on the data, multiple ontologies can be used to “tag” the observed

data with orthogonal concepts at the same time. As a result, the logged data is pre-processed and structured using high-level concepts; consequently, there is no need for extensive and time-consuming post-processing of raw data. Instead, the data can be analyzed directly and in a more efficient way.

In the remainder of this paper, we first point at related work (Section 2). Then, we introduce an example scenario based on a case study that is currently performed (Section 3). Afterwards, we describe our semantic monitoring and analysis approach in more detail (Section 4), and present an implementation (Section 5). Finally, the paper is concluded.

2 Related Work

Uses of remote product monitoring have been reported before [4,5,6,7]. However, these approaches assume information stakeholders capable of programming and willing to use programming paradigms to achieve the sought-after data. In contrast, our approach aims at means to specify observation in a way that is doable by actual stakeholders of the collected information. Besides that, our approach towards product observation emphasizes the integration of observation functionality into the target system by using a software engineering process which is, in our opinion, necessary for widespread use. While previous work [8,9] describes our product observation approach in more detail, this paper focuses on the novel semantic link between observation and analysis.

The idea of using semantics to perform analysis of processes is not new [10,11,12]. Our analysis approach is based on previous work on semantic process mining techniques [13,14]. Process mining techniques can provide valuable insights into a real-life process based on data registered in event logs and have been successfully applied in practice [15]. *Semantic* process mining enhances the analysis by leveraging semantic information [13]. However, previous works do not present any real-life application of the semantic process mining tools. In this paper, we first applied our semantic process mining techniques to analyze processes based on product usage. More related work can be found in our technical report [16].

3 Example Scenario

In the following we want to use a simple example scenario to explain our approach. This example is a simplified version of (but based on) an industrial case study that is currently being performed.

We consider a product that offers a video playback and recommendation function as depicted in Figure 1. In the upper part of the screen one can see the video that is currently played. The video playback can be paused and resumed, and the playback window can be maximized to be displayed in fullscreen mode and brought back to the normal mode. In the lower part of the screen a number of recommendations related to the current video are displayed (using the right or the left arrow more related recommendations can be explored). Any of these recommendations can be viewed in more detail by moving the mouse pointer over it (as can be seen for the right-most recommendation) and selected for playback, after which it is displayed in the upper part of



Fig. 1. Schematic view on the user interface of a video playback and recommendation function of an industrial product in prototype stage

the screen. New recommendations are then retrieved and displayed according to the selected item. Furthermore, the product has a search function that allows to search for video content by name and categories, which is not shown in Figure 1.

We assume that a prototype of this product should be tested by potential end users in a number of different countries. We want to know how people typically navigate this user interface, and whether this differs depending on the cultural context. For example, it would be interesting to know whether users tend to follow the provided recommendations or rather search for video content on a case-by-case basis. Based on this information, the user interface of the product could be improved to best support the most common interaction flows.

4 Approach

Direct product information (i.e. the recording of the actual usage of a system) is potentially of use to a large group of professionals involved in the product development process: knowledge engineers, product managers, requirements engineers, developers, interaction designers, and other information stakeholders can benefit from such information. Note that the members of this group, in the following referred to as *domain experts*, have traditionally only a rather modest influence during some phases of the product creation process. Especially for the development of innovative products, the expertise of such domain experts is needed. These experts are the target users for our approach: initially, they might have a vague understanding about what should be observed in the product to answer open questions, but iteratively it is possible to map issues to observable items within the product, and finally, to obtain comprehensible and reliable information.

In the remainder of this section, we first provide an overview about our product usage monitoring approach (Section 4.1) and then elaborate on the role of ontologies as a semantic link between the different phases of observation and analysis (Section 4.2).

4.1 Overview

Consider Figure 2, which depicts an overview of our approach. The system we propose is a *combination of a logging framework and a process mining tool*. On top of that, one

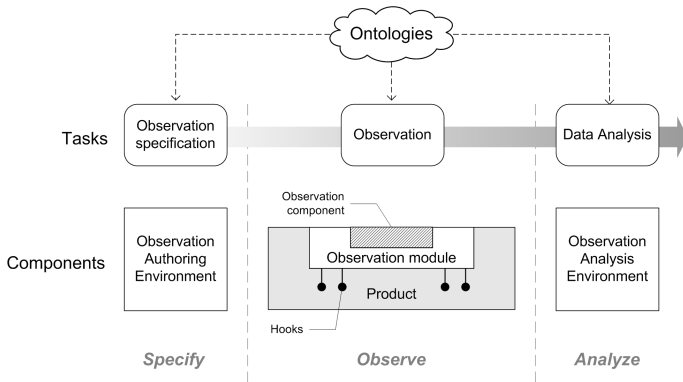


Fig. 2. Overview of our approach towards product usage monitoring and analysis

or more ontologies are used to link collected data items, hence, to connect observation and analysis on the information level. The figure shows that ontologies are connected to all three steps of the flow. Therefore, the definition and maintenance of one or more ontologies should be a concurrent task that accompanies the depicted flow.

In Figure 2 one can see that the product to be observed is equipped with an *observation module* which has access to so-called *hooks*. These hooks and the observation module have to be built into the product beforehand. For example, in the scenario described in Section 3, before actually giving the prototypes to testers at home, the user interface would be instrumented with hooks that are triggered as soon as a video playback is started, a recommendation is selected etc.

During the actual test the following three steps are performed in an iterative manner: (1) The first step of the actual flow is the observation specification: domain experts visually define what information should be observed in the product and how this information relates to the concepts from the ontology. This task is done within an easy, but formal visual language. (2) The outcome are observation specifications which are used to automatically and remotely instruct the observation modules in the various products by simply replacing their *observation component*. The observation modules collect field data during product usage depending on their current configuration and send it to a central data storage. The semantic annotations of the observation specifications enable the observation module to categorize the captured data accordingly on-the-fly. This results in log data with an inherent semantic structure. (3) In the third step (data analysis) the data is processed using various (semantic) process mining techniques which provide different views on the aggregated data. This last step offers the possibility to extract the essence out of a potentially huge data set. Furthermore, it helps to present this information in a comprehensive and directly usable way.

Although the automatic processing chain from observation to analysis consists of several independent parts, it now becomes clear that a common connection is feasible by using ontologies for a semantic content structure. The whole process is of a strongly iterative nature. Cycles between the definition of ontology, observation specification,

observation, and analysis are not only expected but encouraged to finally achieve the most reliable and accurate picture of product usage. For instance, during the observation phase, the domain expert might come across unexpected information that needs a special treatment and the extension of the connected ontology with new concepts. These changes can be carried out directly and lead to an immediate improvement of the quality of collected data.

4.2 Ontologies

Ontologies [3] define the set of shared concepts necessary for the analysis, and formalize their relationships and properties. Ontology elements are organized in a directed graph and there are several formalisms to build ontologies such as OWL [17] and WSM [18]. An example fragment of an ontology is depicted on the right in Figure 3.

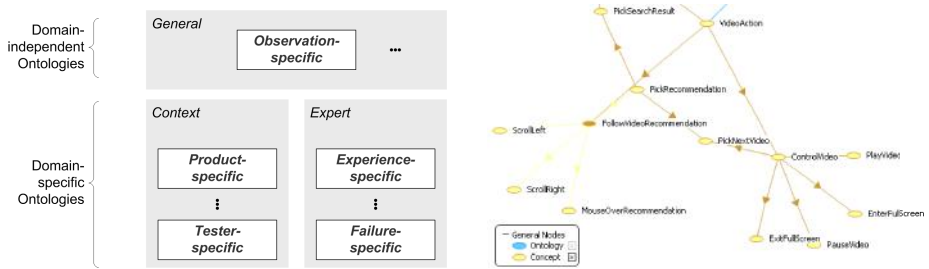


Fig. 3. Types of ontologies relevant for product usage monitoring (left) and an example fragment of a product-specific ontology representing user actions (right)

In the context of our product usage monitoring approach, the ontologies provide the link between conceptual level and information level, i.e., ontology concepts appear in the log data whenever a semantically annotated event is logged. We identify three types of ontologies: *general*, *context* and *expert* (cf. left side in Figure 3). These types can be characterized as follows.

General. General ontologies are domain-independent and they are used to capture concepts that are neither product nor experiment related. They are expected to be highly re-usable for a couple of experiments without changes.

Context. Context ontologies provide information about the setting of an experiment. In other words, they might characterize certain aspects of the product to be observed (i.e., *product-specific*), the habitual context of actual product use, or the people that perform the tests (i.e., *tester-specific*). The applicability of these ontologies may be limited to a certain domain or product group, but they can be re-used across different experiments within that scope.

Expert. Expert ontologies are related to specific analysis purposes. For example, we can think of certain domain-expert views, such as a user experience expert seeking emotional feedback from testers by popping up dialogues on the tested prototypes (i.e., *experience-specific*), or the quality engineer focusing on product failures

(i.e., *failure-specific*). In principle, expert ontologies could be re-used across different product groups.

Note that multiple ontologies are used because the semantic observation and analysis is not done by one person alone. A team of domain experts should be able to work together, and to benefit from each other’s insight into product usage. Therefore, many (potentially orthogonal) views on the topic have to be combined in an efficient way.

Nevertheless, in the remainder of this paper we focus on user actions only. On the right side in Figure 3, an excerpt of a product-specific ontology representing user actions for our example scenario in Section 3 is shown. One can see that concepts are organized in a hierarchical way, i.e., concepts may have one or more superconcepts. For example, the concept ‘PlayVideo’ is a subconcept of the ‘ControlVideo’ category, which in turn is a subconcept of ‘VideoActions’. These subsumption relationships are a very useful tool as they enable the analysis of the data on different levels of abstraction.

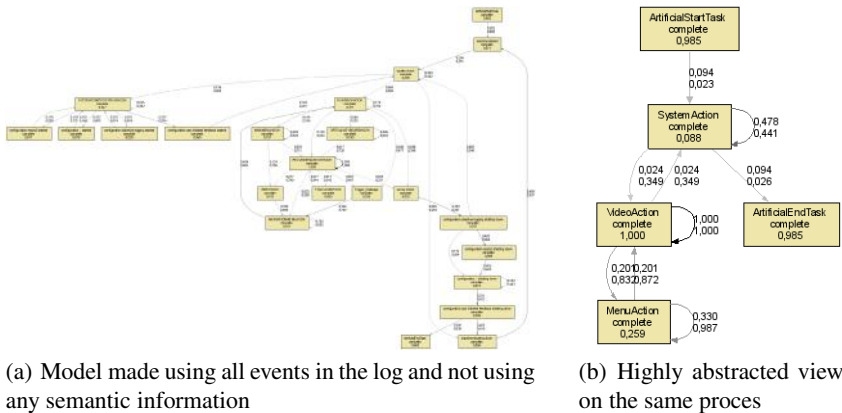


Fig. 4. Two models that were mined from the same log data, but using different abstraction levels

This is illustrated by Figure 4, where process mining was used to automatically create a process model from the data collected in the example scenario. In the model depicted in Figure 4(a) the raw data and no semantic annotations were used to create the model. In fact, this model not only contains steps related to user actions but also includes unrelated information such as status checks of the observation system itself (since these are logged as well). In contrast, the model in Figure 4(b) only contains process steps relating to user actions. Furthermore, the depicted model provides a highly abstract view by making use of the semantic information in the log data. For example, since both ‘PlayVideo’ and ‘PauseVideo’ are a ‘VideoAction’ according to our ontology, they are not differentiated in this model. Note that although the model depicted in Figure 4(b) may seem too general, the level of abstraction can be varied at wish and without the need to modify the actual data itself. This way, varying models with even heterogeneous degrees of abstraction can be created easily. For example, we can create a model that provides a detailed view on ‘VideoActions’ but fully abstracts from ‘MenuActions’.

5 Implementation

We have fully implemented the approach outlined above and are currently testing it in an industrial case study. In the following two sub sections, we describe the tools that we used for the realization (D’PUIS and ProM), focussing on newly added functionality and the semantic aspects.

5.1 Observation Specification and Data Collection via D’PUIS

We have developed the D’PUIS (Dynamic Product Usage Information System) [89] as a platform-specific realization of the specification and observation approach depicted in Figure 2. This system consists of the following parts: (i) a visual editor to create observation specifications, (ii) a web application that distributes observation specifications as observation components and provides storage for collected product data, and (iii) an observation module which is integrated into product instances. An infrastructure connects these parts and enables an automatic flow from observation specification to actual product usage data.

In the context of semantically supported data collection, an interesting part of the observation system is the visual language as the place where semantic links between data items are initially constructed. To do this, the visual language was extended to automatically incorporate each concept from a linked ontology as an available *Semantic Hook*. If such a semantic hook is triggered, a semantically annotated log entry is created. Often, the actual platform hooks can be connected to semantic hooks in a straightforward way, merely differentiating between a number of options. However, the processing nodes of our visual language also allow for more powerful observation specifications, which is demonstrated by the following example.

Consider Figure 5, which depicts a part of the observation specification for our example scenario in the visual editor. The lightbrown block in the middle represents the ‘RecommendationHook’ that is triggered whenever a recommended video is selected for playback by the user (cf. Section 3). However, in fact the same user interface component (and, thus, the same platform hook) is triggered when a user picks a search result after explicitly searching for video content. But in our scenario we want to differentiate between these two conceptual actions. Fortunately, we can create a context-aware observation specification that only triggers the semantic hook ‘PickNextVideo’ (i.e., the actual recommendation) when the user did not just enter the search mode via checking for the context node ‘EnterSearch’, which is also based on semantic information. If the search mode was entered before, the semantic hook ‘PickSearchResult’ is triggered instead. Note that this kind of domain-dependent reasoning would normally need to be made later in the analysis stage, or hard-coded into the product.

Data that is acquired in the described way is not only more meaningful, but also it is *self-contained*. This is an important step forward as all the (usually implicit) information about the observation process, such as the characteristics of the observation environment, and the nature of data sources, is *explicitly* stated in a *machine-readable form*. In the analysis phase, specialized semantic process mining techniques can then exploit such information efficiently.

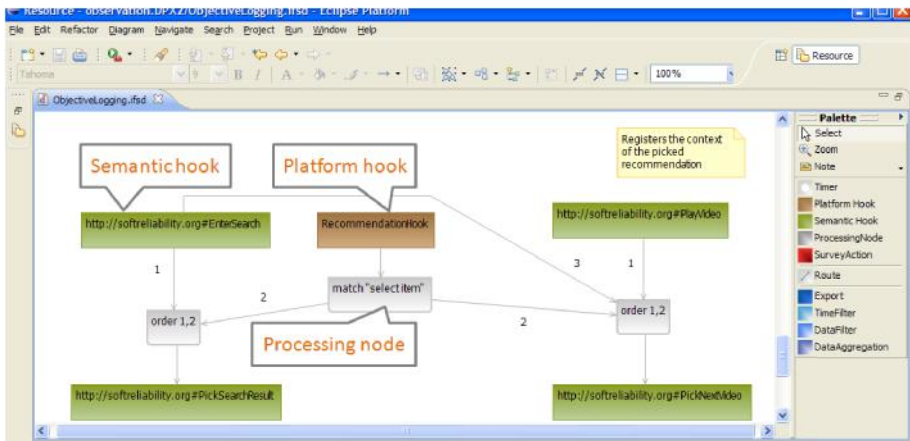


Fig. 5. Visual editor for observation specification with an example specification from the example scenario

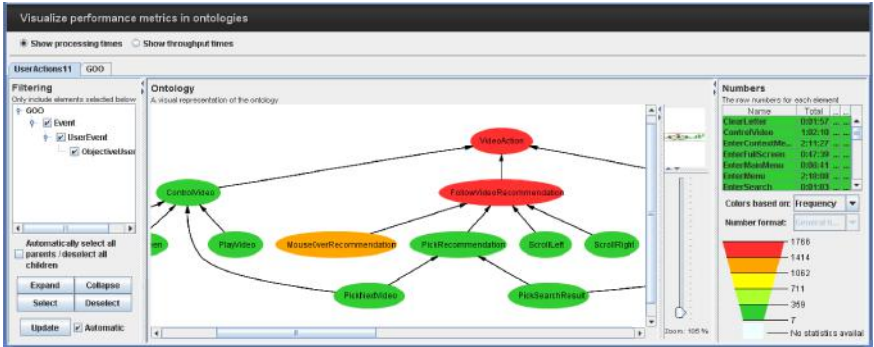
5.2 Semantic Process Mining Using ProM

To be able to analyze the log data with our process mining tool kit ProM [19], we have developed a *ProMimport* [20] plug-in that automatically extracts the recorded data from the D’PUIS database and converts them to the SA-MXML (*Semantically Annotated Mining XML*) format [14]. Note that this data conversion preserves the semantic annotations collected during the observation phase for analysis. Process mining techniques support various types of analysis based on the behavior registered during the execution of some process [15]. Semantic process mining uses semantic information to lift the analysis provided by current process mining techniques to the conceptual level [13,14]. Seven semantic process mining plug-ins have been added to the ProM tool so far; we briefly introduce the following two: Performance Metrics in Ontologies and the Ontology Abstraction Filter.

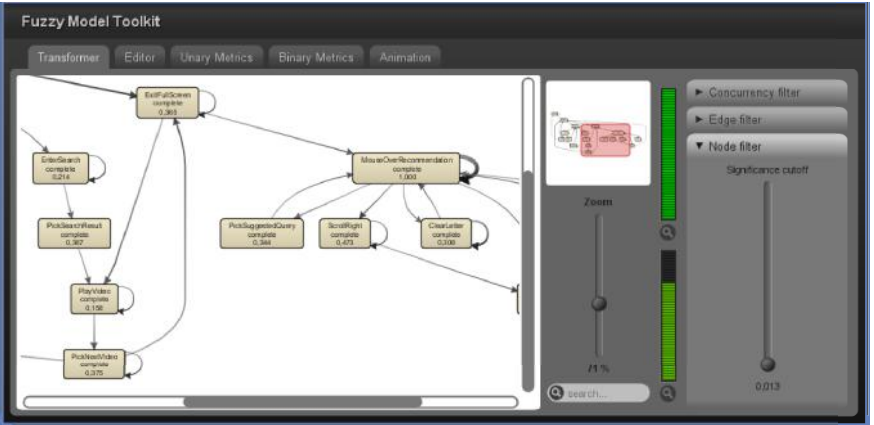
The *Performance Metrics in Ontologies* plug-in provides feedback about (i) the processing times of tasks (or events) and (ii) throughput times of process executions. In our approach, the feedback in (i) is particularly important because it indicates how much time users typically spend in using certain functionalities of products. Moreover, this plug-in also shows how frequently instances of a given concept have been performed. Figure 6(a) contains a screenshot of this plug-in in action. Note that the coloring of the concepts in the ontology is based on the frequency of instances. From this graph, it is very intuitive to spot that the users in our example scenario were more often navigating between recommendations (concept ‘FollowVideoRecommendation’) than actually playing videos (concept ‘ControlVideo’).

The *Ontology Abstraction Filter* plug-in supports ontology-based run time filtering of the data in a way that is accessible to all existing process mining algorithms in ProM (also if they are unaware of the semantic annotations in the log). In this filter, the desired

¹ Both ProM and ProMimport are open-source and freely available at www.processmining.org.



(a) Screenshot of the *Performance Metrics in Ontologies* semantic ProM plug-in. The current view shows the frequencies of tasks linking to concepts



(b) Screenshot of the *Fuzzy Miner* plug-in in ProM. Before, the semantic information in the log has been used to filter only events referring to user actions

Fig. 6. The converted log can be loaded and analyzed using the ProM tool

level of abstraction is determined by selecting or deselecting concepts linked to events (the actual instances of these concepts) in logs. Afterwards, process mining algorithms can be used to create models on the current level of abstraction. For example, Figure 6(b) depicts a screenshot of the Fuzzy Miner [21] showing a detailed process model of the user actions. One can see that after searching for a video (‘EnterSearch’ followed by ‘PickSearchResult’ and ‘PlayVideo’) users tend to follow recommendations (‘PickNextVideo’) rather than going back to explicitly search for further videos.

6 Conclusion

In this paper, we presented a novel approach to semantically link the observation and analysis of product usage data by conceptual information captured in ontologies. This link renders a potential mass of captured data items more manageable, and reduces the

need for extensive post-processing. The presented approach ensures high information quality and speeds up the information feedback cycle towards development. Furthermore, we presented a tool chain that supports our approach throughout the phases of observation specification, data collection, and analysis. This chain of connected data processing components offers also the flexibility to change observation remotely on-the-fly, enabling fast data collection and analysis iterations.

Our vision is a fully automated data collection, processing, analysis and presentation chain which is specified by only a few (potentially re-usable) documents. Ontologies and visual languages seem to be good candidates for such specification documents as they are accessible to the actual stakeholders of the observed usage data (e.g., the various domain experts). By putting these people in the position of being able to *specify what they want to observe*, one of the main problems in log analysis, namely data quality, can be addressed. In many real-life scenarios, the data are often still of a poor quality; because of a low priority in implementing logging facilities, and a lack of anticipation of the kind of analysis that should be eventually performed, collected data are not good enough to answer all the questions of interest. However, due to the immense opportunities and increasing feasibility (resulting from novel automated approaches as presented in this paper) it can be expected, that the integration of observation functionality will have a more prominent role in future product developments. As a consequence, better analysis results can be expected.

Acknowledgements. This work is being sponsored by the Dutch Ministry of Economic Affairs under the IOP-IPCR program. Some of the authors are also supported by the European project SUPER. Furthermore, the authors would like to thank the industrial team for the possibility of applying our approach in a real product development context.

References

1. Brombacher, A., Sander, P., Sonnemans, P., Rouvroye, J.: Managing product reliability in business processes 'under pressure'. *Reliability Engineering & System Safety* 88, 137–146 (2005)
2. den Bouwmeester, K., Bosma, E.: Phases of use: a means to identify factors that influence product utilization. In: *CHI 2006: CHI 2006 extended abstracts on Human factors in computing systems*, pp. 117–122. ACM Press, New York (2006)
3. Gruber, T.: A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition* 5(2), 199–220 (1993)
4. Hartson, H., Castillo, J.: Remote evaluation for post-deployment usability improvement. In: *Proceedings of the working conference on Advanced visual interfaces*, pp. 22–29 (1998)
5. Hilbert, D.M., Redmiles, D.F.: An approach to large-scale collection of application usage data over the internet. In: *ICSE 1998*, pp. 136–145 (1998)
6. Kabitzsch, K., Vasyutynskyy, V.: Architecture and data model for monitoring of distributed automation systems. In: *1st IFAC Symposium on Telematics Applications In Automation and Robotics*, Helsinki (2004)
7. Shifroni, E., Shanon, B.: Interactive user modeling: An integrative explicit-implicit approach. *User Modeling and User-Adapted Interaction* 2(4), 331–365 (1992)

8. Funk, M., van der Putten, P.H.A., Corporaal, H.: Specification for user modeling with self-observing systems. In: Proceedings of the First International Conference on Advances in Computer-Human Interaction, Saint Luce, Martinique, IARIA, pp. 243–248. IEEE Computer Society, Los Alamitos (2008)
9. Funk, M., van der Putten, P.H.A., Corporaal, H.: UML profile for modeling product observation. In: Proceedings of the Forum on Specification and Design Languages (FDL 2008), Stuttgart, Germany, pp. 185–190. ECSI, IEEE Computer Society (2008) ISBN: 978-1-4244-2264-7
10. Casati, F., Shan, M.: Semantic Analysis of Business Process Executions. In: Jensen, C.S., Jeffery, K., Pokorný, J., Šaltenis, S., Bertino, E., Böhm, K., Jarke, M. (eds.) EDBT 2002. LNCS, vol. 2287, pp. 287–296. Springer, Heidelberg (2002)
11. Hepp, M., Leymann, F., Domingue, J., Wahler, A., Fensel, D.: Semantic Business Process Management: a Vision Towards Using Semantic Web services for Business Process Management. In: IEEE International Conference on e-Business Engineering (ICEBE 2005), pp. 535–540 (2005)
12. O’Riain, S., Spyns, P.: Enhancing the Business Analysis Function with Semantics. In: Meersman, R., Tari, Z. (eds.) OTM 2006. LNCS, vol. 4275, pp. 818–835. Springer, Heidelberg (2006)
13. Alves de Medeiros, A., Pedrinaci, C., van der Aalst, W., Domingue, J., Song, M., Rozinat, A., Norton, B., Cabral, L.: An Outlook on Semantic Business Process Mining and Monitoring. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2007, Part II. LNCS, vol. 4806, pp. 1244–1255. Springer, Heidelberg (2007)
14. Alves de Medeiros, A.K., van der Aalst, W.M.P., Pedrinaci, C.: Semantic Process Mining Tools: Core Building Blocks. In: Proceedings of the 16th European Conference on Information Systems, ECIS (2008)
15. van der Aalst, W., Reijers, H., Weijters, A., van Dongen, B., Alves de Medeiros, A., Song, M., Verbeek, H.: Business Process Mining: An Industrial Application. Information Systems 32(5), 713–732 (2007)
16. Funk, M., Rozinat, A., Alves de Medeiros, A., van der Putten, P., Corporaal, H., van der Aalst, W.: Semantic concepts in product usage monitoring and analysis. Technical Report ESR-2008-10, Eindhoven University of Technology (2008)
17. W3C: Web Ontology Language (OWL), <http://www.w3.org/2004/OWL/>
18. de Bruijn, J., Lausen, H., Polleres, A., Fensel, D.: The web service modeling language wsml: An overview. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 590–604. Springer, Heidelberg (2006)
19. van der Aalst, W.M.P., van Dongen, B.F., Günther, C.W., Mans, R.S., Alves de Medeiros, A.K., Rozinat, A., Rubin, V., Song, M., Verbeek, H.M.W., Weijters, A.J.M.M.: ProM 4.0: Comprehensive Support for Real Process Analysis. In: Kleijn, J., Yakovlev, A. (eds.) ICATPN 2007. LNCS, vol. 4546, pp. 484–494. Springer, Heidelberg (2007)
20. Günther, C.W., van der Aalst, W.M.P.: A Generic Import Framework for Process Event Logs. In: Eder, J., Dustdar, S. (eds.) BPM Workshops 2006. LNCS, vol. 4103, pp. 81–92. Springer, Heidelberg (2006)
21. Günther, C., Aalst, W.: Fuzzy Mining: Adaptive Process Simplification Based on Multi-perspective Metrics. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 328–343. Springer, Heidelberg (2007)

Using GA and KMP Algorithm to Implement an Approach to Learning Through Intelligent Framework Documentation

Hajar Mat Jani¹ and Sai Peck Lee²

¹ College of Information Technology, Universiti Tenaga Nasional
Km. 7, Jalan Kajang-Puchong, 43009 Kajang, Selangor, Malaysia
hajar@uniten.edu.my

² Faculty of Computer Science and Information Technology
University Malaya, 50603 Kuala Lumpur, Malaysia
saipeck@um.edu.my

Abstract. Object-oriented application framework is one of the most important implementations of object-oriented software engineering. Normally, a user takes several months of learning in order to become highly productive in using a specific object-oriented application framework. Without proper documentation, frameworks are not very usable to framework users. Currently available framework documentation approaches are not very effective for new framework users, and this scenario tends to discourage new users in using frameworks. The main objective of this paper is to propose and implement an intelligent framework documentation approach that integrates case-based learning (CBL) with genetic algorithm (GA) and Knuth-Morris-Pratt (KMP) pattern matching algorithm with the intention of making learning a framework more effective. GA assists in optimizing the search process and performs machine learning. Within the GA, nearest neighbor algorithm is used in determining the most similar recorded case that can be used in solving the new case. A new case is retained in the case base for future retrievals. A framework user is allowed to select from a list of features provided by the framework that he or she is interested in learning, and the system will give an example of application related to the selected features. This paper concludes with a prototype that implements the intelligent framework documentation approach.

Keywords: Framework documentation, genetic algorithm (GA), Knuth-Morris-Pratt (KMP) pattern matching algorithm.

1 Introduction

An object-oriented application framework's main objective is to allow the reuse of both design and code in the development of new applications using the framework. A framework sets a frame of mind for solving problems and provides means to realize solutions in software within a short period of time. It implements the software architecture for a family of applications with similar characteristics, which are derived by specialization through application-specific code [1]. In order for a framework to be usable, good and effective documentation is required, especially for new framework users.

Documentation can be viewed as a formal way of communicating knowledge to users, and if the users have difficulty in understanding the documentation, the documentation is said to be ineffective. Therefore, each user of a framework must be able to easily understand the framework simply by performing simple and effective learning using the provided documentation.

2 Objective

The main objective of this paper is to propose and implement an intelligent framework documentation approach that combines genetic algorithm (GA) and Knuth-Morris-Pratt (KMP) pattern matching algorithm with the intention of making learning more effective.

To make the approach intelligent, the concept of machine learning through evolutionary learning is performed using the GA to improve solutions over time when searching for the most optimal solution. In addition, a new case solution is derived or adapted from existing case or set of cases using the combination of GA and KMP algorithm. Furthermore, solutions that differ from existing cases are saved in the knowledge base for future retrievals.

3 Framework Documentation

Documentation plays a very crucial role in learning how to use an object-oriented application framework. It is a known fact that the main objective of using a framework is to reduce the amount of time to develop application software.

Basically, there are two main objectives of framework documentation. First, information about framework design and other related information need to be made available and communicated during the framework development. Second, information on how to use the framework must be easily available to the user and must be delivered along with the framework [2].

3.1 Documentation Approaches

Currently, a number of approaches are being used for documenting frameworks, and from our previous study [3], we managed to identify the following documentation approaches: tutorials, cookbooks and recipes, design patterns, frequently asked questions, framework description language (FDL), contracts, run-time errors documents, web approach, formal specification, hooks model, and case-based reasoning.

Most of the available documentation approaches are not very user-friendly to new users. This is verified based on our survey [3] from which majority of users agreed that the documentation approaches are not effective for new framework users. Furthermore, they agreed that an intelligent documentation approach with machine learning ability should be introduced. The users also suggested that examples of complete programs are required if we want to enhance learning and this will definitely make the documentation more usable.

3.2 Literature Review

In a paper entitled “Minimalist Documentation of Frameworks”, [4] Østerbye discussed a minimalist approach that combined reference manuals, examples, tutorials, and run-time error messages into a hypertext-based documentation of frameworks that used the minimalist documentation technique.

Ian Chai [5] in his thesis performed an empirical study on three documentation approaches: traditional step-by-step instructions, minimalist documentation, and patterns. He performed the experiment on fifteen subjects, and the results showed that the subjects using the minimalist documentation finished faster than the ones using the other two approaches [5]. The only limitation of the study performed here was that the researcher just compared the completion time of the subjects when using the specified approach.

Johnson [6] discussed the use of patterns in framework documentation. He adopted several important features of earlier pattern languages, but modified the format of presenting the patterns to make them more appealing. He emphasized the fact that patterns should be problem-oriented, not solution-oriented. Johnson described the HotDraw framework using several patterns, where each pattern gave the description of how to solve a small portion of a larger design problem. In his paper, Johnson concluded that patterns are very useful in teaching new (first-timer) framework users on how to use a particular framework.

Froehlich, Hoover, Liu, and Sorenson suggested the use of hooks to augment application framework documentation in their paper entitled “Hooking into Object-Oriented Application Frameworks [7].” The main objective of their study was to introduce the concept of hooks as a means of documenting and providing guidance on the intended use of a framework. Hooks are defined by the framework builder, who is the most knowledgeable about the framework with the intention of passing on his or her knowledge to the application developer. The researchers further demonstrated the use of hooks by implementing hooks to the HotDraw application framework documentation.

CBR framework documentation approach was proposed in a paper written by Gomez-Albarran, Gonzalez-Calero, and Fernandez-Chamizo [8]. The CBR approach introduced by these researchers is only to complement existing documentation methods and at the same time faces their drawbacks.

Chen and Xu [9] discussed framework documentation issues in their paper. They discussed the need to have multiple views of a framework, but noted that most of the time having all possible views of a framework is not practical and very costly. Chen and Xu also proposed the heuristics of framework documentation and concluded that by using those heuristics properly, good documentation can be produced.

With regards to learning, based on a study conducted in [10], the authors discovered that both users’ learning styles and choice of colors are important in ensuring a system’s effectiveness, user-friendliness, and usability. The results of the survey indicated that most respondents relate the usability of a system with individual’s learning style.

4 Implementation of GA and KMP

Two main algorithms that are used in the proposed approach are genetic algorithm (GA) and Knuth-Morris-Pratt (KMP) pattern matching algorithm. The following sub-sections describe how these two algorithms play important roles during the machine learning process.

4.1 Genetic Algorithm (GA)

John Holland [11] introduced the concept of genetic algorithms in the early 1970s. In general, “Genetic Algorithms (GAs) are adaptive heuristic search algorithms premised on the evolutionary ideas of natural selection and genetics [12].” GA is

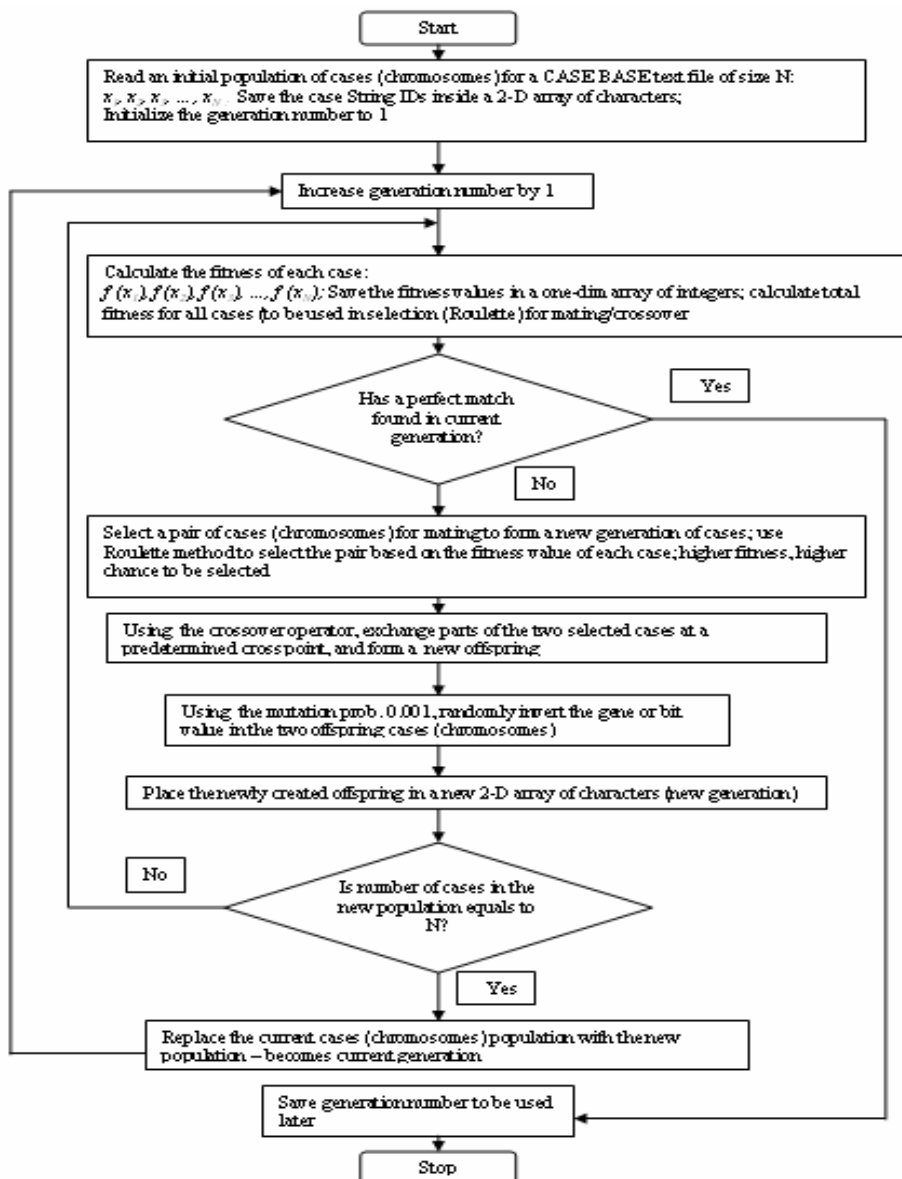


Fig. 1. The GA in the proposed intelligent framework documentation

based on the basic concept to simulate processes in the natural environment necessary for evolution. It holds the principles laid down by Charles Darwin of “survival of the fittest.” GA is mainly used when performing a random search within a large search space, and is widely regarded as an algorithm that can optimize the search process.

In our approach, GA actually plays two roles; first, as an optimizer during the search for the most optimal (best) solution for a case, and second, as the basis for machine learning [12]. In this approach, GA plays the role of identifying the case that is most similar to the current case. If an organism (a case) has a way of surviving within a population over successive generations, then it can be said that this organism is capable of learning to predict changes in its environment [11]. This is an evolutionary approach to machine learning, which is based on computational models of natural selection and genetics.

Within our approach, machine learning occurs when the system learns from past experiences by modifying existing most similar case(s) to find the best solution to the new case at hand. Learning here means, the system can use the newly learned case in the future since the new case is stored in the case base for future retrievals.

The following gives a diagram (Fig. 1) that shows how the GA [11] is applied in our approach.

4.2 Knuth-Morris-Pratt (KMP) Algorithm

KMP pattern matching algorithm is used during the adaptation and learning process of the new problem. Basically, the KMP algorithm [13] searches for the occurrences

Algorithm *kmp_table*:

Input:

An array of characters, W (the word to be analyzed)

An array of integers, T (the table to be filled)

Output:

None (but during operation, it populates the table)

Define variables:

An integer, pos $\leftarrow$ 2 (the current position we are computing in T)

An integer, cnd $\leftarrow$ 0 (the zero-based index in W of the next character of the current candidate substring)

(the first few values are fixed but different from what the algorithm might suggest)

let T[0] $\leftarrow$ -1, T[1] $\leftarrow$ 0

while pos is less than the length of W, **do**:

(First case: the substring continues)

if W[pos - 1] = W[cnd], **let** T[pos] $\leftarrow$ cnd + 1,
pos $\leftarrow$ pos + 1, cnd $\leftarrow$ cnd + 1

(Second case: it doesn't, but we can fall back)

otherwise, if cnd > 0, **let** cnd $\leftarrow$ T[cnd]

(Third case: we have run out of candidates. Note cnd = 0)

otherwise, let T[pos] $\leftarrow$ 0, pos $\leftarrow$ pos + 1

Fig. 2. KMP pattern matching algorithm's “partial match” table

of a specified string of “word” W within a given line of “text string” S by employing the observation that when a mismatch occurs, the word itself contains sufficient information to decide where the next match could begin, and thus bypassing re-examination of previously matched characters. This will increase the efficiency of the pattern matching process.

The KMP algorithm was introduced by Donald Knuth and Vaughan Pratt, and at the same time independently by J.H. Morris in 1977. A “partial match” table T of previously matched characters within the “text string” is kept, and this table is used in setting the indexes for the searched “word” W , and the line of text being searched, S .

The general algorithm for computing the “partial match” table is given in Fig. 2 [13]. During the implementation phase of the proposed approach, the KMP pattern matching algorithm is modified as appropriate in order to suit the problem domain, and also to make the algorithm more efficient.

5 The Intelligent Documentation Approach

Basically, the approach uses examples of past framework usage experiences and cases in order to solve new cases or problems. A case represents a problem situation, while a past case is a previously “experienced situation” that has been retained and learned so that it can be reused in solving future problems [14]. Learning by remembering previously recorded experiences or cases is normally referred to as *case-based learning* (CBL). Case-based learning is simply learning by recalling similar cases that had happened in the past that are stored in the memory of a system in order to solve the current problem at hand. Actually, CBL is the result of learning when using *case-based reasoning* (CBR) [15]. CBR promotes learning from experiences. The term CBL is sometimes also referred to as example-based learning [14], and that’s the reason why we need to have complete examples to represent past cases before we can start the reasoning process using CBR. We will implement CBL using the above-mentioned algorithms, which are GA and KMP. The main use of the GA is to optimize searching process of the needed case solution, which must be the best solution (perfect match), or the most similar solution, and this solution must be adapted to suit the new case requirements.

The GA also assists in learning because the chromosomes within a GA are actually representing past recorded cases. In other words, the case-based learning is done through the use of GA. As described earlier, the initial population of the GA is the list of stored cases (chromosomes). At each GA generation, the fitness value of a case is calculated and used as the selection criterion for being included in the next generation. The fitness function simply measures how close the searched case (chromosome) to the desired case or “model chromosome”. Here, the *nearest neighbor algorithm* is used to find the most similar case. The selection process is done using the *Roulette* method, where cases with higher fitness values will have higher probability to be selected to be included in the next generation. This is to ensure better average of fitness values as the generation progresses, so that the cases (chromosomes) that form the new generation have better solutions as compared to previous generations. The final generation is the generation that contains a perfect match. The perfect match is either in the case base or must be constructed by adaptation using the most similar case or set of cases.

During the adaptation process, KMP pattern matching is used to find certain patterns within the cases that are used during the adaptation. We need to replace certain patterns with new patterns to ensure accurate substitution and transformation of the most similar case solution into the new case solution.

The application of GA along with CBL/CBR and KMP algorithm will result in better quality and intelligent framework documentation. In fact, the combination of GA and CBR approach has been used in other domains such as in constructing a personalized e-learning system [16], and in a decision support system for housing customization [17]. In both systems, CBR/CBL was used in remembering and reusing past cases or experiences in the respective problem domain, and this directly enhanced the learning ability of both the system and the users.

6 Implementation: The Prototype

A prototype is developed to present the proposed approach. We developed the prototype using Java, and used Java Swing GUI components [18] as an example of a framework.

Basically, the proposed system works as follows [19]:

1. User enters a query/problem/case.
2. The system interprets the query by identifying the main features of the query that results in some feature values (problem description) of the new case. This is represented as a case string ID, and is considered as an index for retrieving the case later.
3. Based on the feature values, the desired (the most similar) case or set of candidate cases will be *retrieved* from the knowledge base (the case base library of files).
4. If a perfect match is found, then the case is *reused* as it is. Genetic algorithm (GA) is used in optimizing the search process.
5. If there is no perfect match, the most similar case or set of cases has to be retrieved, and will be *revised* or adapted to suit the new case/problem. Here, both GA and Knuth-Morris-Pratt (KMP) pattern matching algorithm are used to perform adaptation.
6. The documentation system learns the new case and *retains* it in the case base for future use. *Learning* occurs when the case base is updated, and the new case is saved inside the case base file information and the new case solution is saved in the library of cases (files).
7. Finally, the system displays the proposed solution to the user.

In this prototype, we treat the core component of the system as a case-based reasoner, which actually uses case-based reasoning artificial intelligence technique to perform reasoning by remembering past cases or experiences. These experiences are stored inside a knowledge base that consists of a library of files or cases related to the system. Each Java file is a case, which contains a complete solution for a particular problem or query.

The collaboration diagram and the sample test runs are given in the following sub-sections.

6.1 Collaboration Diagram

A collaboration diagram models objects (instances of classes) and their sequenced communication between each other. We use the unnamed, classified instance type to represent all objects within the collaboration diagram (Fig. 3). Only the class name (underlined) preceded by a colon is stated. The main purpose of the collaboration is to show interactions among objects along with the messages sent among them. These

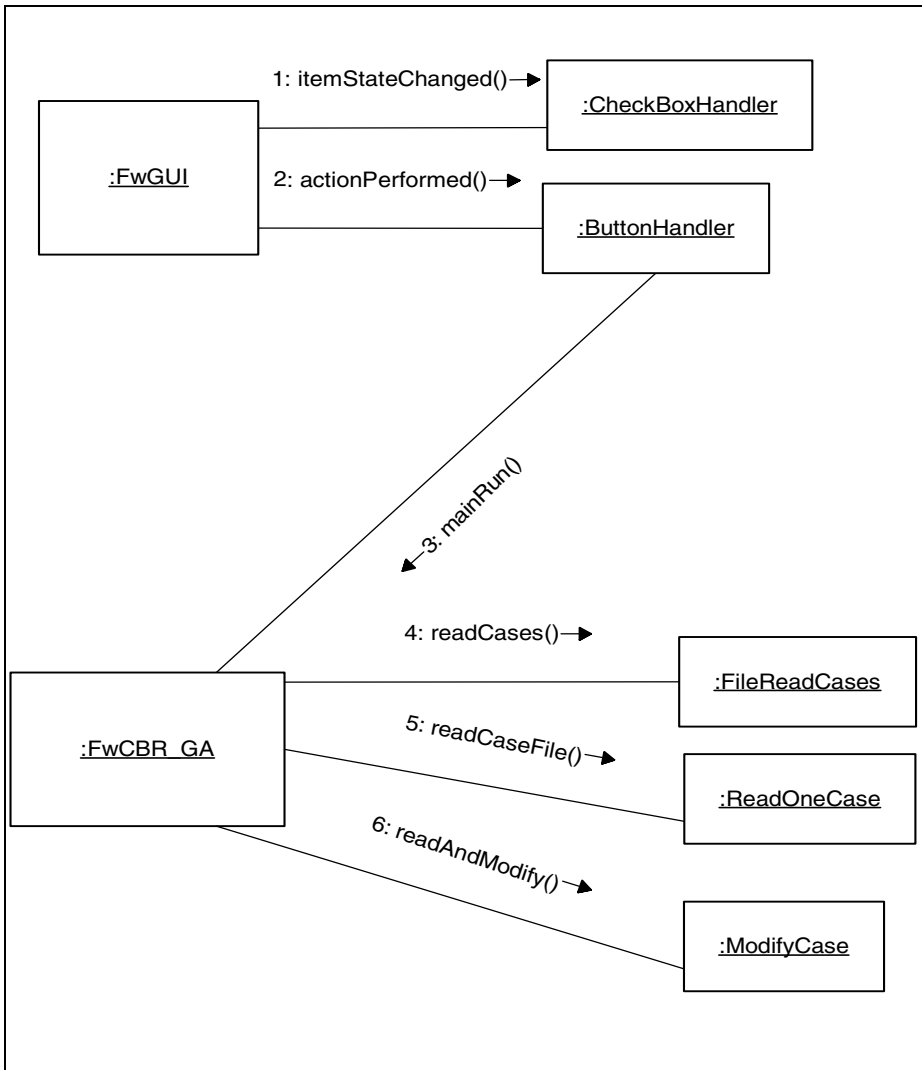


Fig. 3. The collaboration diagram that models the system

messages, written on or next to the respective association, are the means by which objects communicate and inform other objects what need to be accomplished by the objects receiving the messages.

The message sent by one class to another class is accompanied by a number, which is written right before the message name (method name). The numbers indicate the sequence in which the messages are sent to accomplish all required tasks in meeting the requirements of the system. Class *FwGUI* is responsible for getting and interpreting the new case, while class *FwCBR_GA* is responsible for retrieving and processing the cases using CBR/CBL, GA and KMP algorithm in order to come up with a correct solution to the new case.

6.2 User Interface and Sample Results

The following two screen captures are sample user interfaces when the prototype is executed. The first screen capture (Fig. 4) gives the input screen where a framework

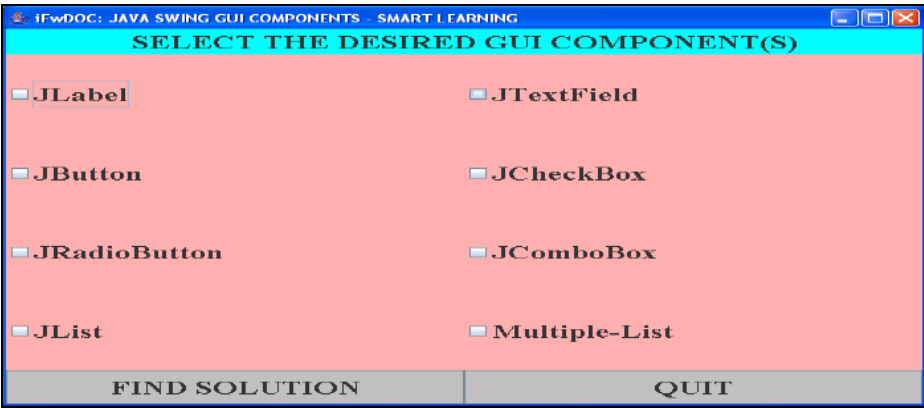


Fig. 4. Sample user interface to get the input from the framework user

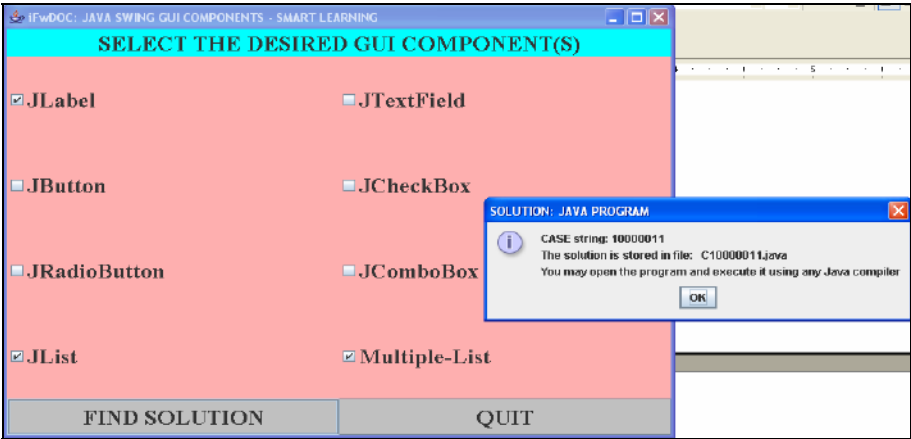


Fig. 5. Sample displayed message after a case has been processed

user can enter his or her choices of the Swing GUI components that he or she would like to learn about. The second screen capture (Fig. 5) shows the dialog box that contains the message regarding where the solution (Java program) to the case is saved. In this example, the user selected three GUI components and a perfect match was not found. The most similar case was adapted to suit the new case, and a new Java file was created and saved in the library of files, which is part of the system's knowledge base (memory). The case string ID is used in constructing the name of the Java program so that in the future the file can be retrieved based on this ID.

7 Discussions

Several observations have been made after the approach has been implemented using a simple prototype, which include:

- The user interface for implementing the approach is simple enough to make sure that users can get to the bottom of the problem without having to waste a lot of time. Furthermore, the user is able to continue entering new data for new cases without having to move away from the main frame. This makes the framework to be more *user-friendly* and more *usable*.
- The system is able to *adapt* cases correctly and new cases information and solutions are learned by the system by saving them for future retrievals. So, machine learning is present within the system.
- The processing time taken by the system is almost as fast as a click of a button, and users can immediately see the results. This shortens the time taken to learn using the framework, and this increases *efficiency*.

Based on the overall results, the approach is successful in implementing learning through intelligent framework documentation. The only limitation of this approach is that it still requires users to refer to other types of documentation for solving problems that are complex. This approach is only meant to be used by new framework users with no prior experience in using the framework.

8 Conclusion and Future Work

In conclusion, the intelligent approach to documenting frameworks can be seen as a new way of promoting the use of the frameworks. Users will be more interested in using a newly introduced framework if the documentation is more *user-friendly* and more *effective* to new framework users. This will also enhance the intended *usability* of the framework.

We suggest that in the future the cases be categorized into simple, moderate, and complex examples, so that users can learn to use the components in many different ways. Moderate and advanced framework users will also be able to benefit from this

documentation approach. It is hoped that this approach can also be applied for documenting other types of software.

References

1. Fayad, M.E.: E-Frame: A Process-based Object-Oriented Framework for E-Commerce. University of Nebraska at Lincoln, Nebraska, United States of America, <http://www-engr.sjsu.edu/~fayad/publications/conference/E-frame.doc>
2. Fayad, M.E., Schmidt, D.C., Johnson, R.E.: Building Application Frameworks. Wiley, Canada (1999)
3. Mat Jani, H., Lee, S.P.: A Study on Object-Oriented Application Frameworks Documentation: Documenting Approaches. In: Proc. Informatics & RWICT 2004 International Conference, vol. 1, pp. 383–400 (2004)
4. Østerbye, K.: Minimalist Documentation of Frameworks (1999), <http://www.literateprogramming.com/minimal99.pdf>
5. Chai, I.: Pedagogical Framework Documentation: How to Document Object-Oriented Frameworks, an Empirical Study. Ph.D Thesis, University of Illinois at Urbana-Champaign (2000)
6. Johnson, R.E.: Documenting Frameworks using Patterns. In: Proc. OOPSLA 1992 Conf., pp. 63–76 (1992)
7. Froehlich, G., Hoover, H.J., Liu, L., Sorenson, P.: Hooking into Object-Oriented Application Frameworks. In: Proc. 19th Int'l Conf. Software Engineering, pp. 491–501 (1997)
8. Gomez-Albarran, M., Gonzalez-Calero, P., Fernandez-Chamizo, C.: Profiting from Case-based Reasoning in Framework Documentation. IEEE Software Engineering Journal, 111–122 (2001)
9. Chen, Q., Xu, F.: Framework Issue: Framework Documentation. Department of Computer Science. University of Nevada at Reno, http://www.cse.unr.edu/~chen_q/docu.html
10. Zakrezewska, D., Wojciechowski, A.: Identifying Students Usability Needs in Collaborative Learning Environments. In: Proc. 2008 Conference on Human System Interaction (HSI 2008), pp. 862–867. IEEE/UITM, Poland (2008)
11. Negnevitsky, M.: Artificial Intelligence: A Guide to Intelligent Systems. Addison-Wesley, Pearson Education Limited (2002)
12. Genetic Algorithms, http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/tcw2/report.html
13. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 2nd edn. MIT Press and McGraw-Hill, Cambridge (2001), http://en.wikipedia.org/wiki/Knuth-Morris-Pratt_algorithm
14. Aamodt, A., Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations and System Approaches. AI Communications 7(1), 39–59 (1994)
15. Leake, D.B.: CBR in Context: The Present and Future: Cited In: Case-Based Reasoning: Experiences. In: Leake, D.B. (ed.) Lessons & Future Directions, pp. 3–30. AAAI Press/The MIT Press (1996)
16. Huang, M.J., Huang, H.S., Chen, M.Y.: Constructing a Personalized E-Learning System Based on Genetic Algorithm and Case-Based Reasoning Approach. Expert Systems with Applications Elsevier J. (2006)

17. Juan, Y.K., Shih, S.G., Perng, Y.H.: Decision Support For Housing Customization: A Hybrid Approach using Case-Based Reasoning and Genetic Algorithm. *Expert Systems with Applications Elsevier J.*, 83–93 (2006)
18. Deitel, H.M., Deitel, P.J.: *Java How to Program*, 4th edn. Prentice Hall, New Jersey (2002)
19. Mat Jani, H., Lee, S.P.: Applying Case Reuse and Rule-Based Reasoning in Object-Oriented Application Framework: Analysis and Design. In: *Proc. 2008 Conference on Human System Interaction (HSI 2008)*, pp. 597–602. IEEE/UITM, Poland (2008)

Computer-Based Assessment: From Objective Tests to Automated Essay Grading. Now for Automated Essay Writing?

Robert Williams¹ and John Nash²

¹ Curtin University of Technology, Perth, Western Australia, Australia

² University of Ottawa, Ottawa, Ontario, Canada

bob.williams@cbs.curtin.edu.au, nashjc@uottawa.ca

Abstract. Assessment of student learning is an important task undertaken by educators. However it can be time consuming and costly for humans to grade student work. Technology has been available to assist teachers in grading objective tests for several decades; however these true-false and multiple choice tests do not capture the deeper aspects of student learning. Essay writing can be used to assess this deeper learning, which includes a student's ability to synthesize his/her thoughts, and argue for propositions. Automated essay grading systems are now starting to be used in the educational sector with some success. They can reduce the cost of grading, and they also eliminate the inconsistencies that are found amongst human graders when marking the same essay. The next development in essay processing technology is automated essay writing. This development will present a new set of challenges for educators. The detection of automatically generated essays may be difficult, and students may be given credit for writing which does not reflect their true ability. An understanding of how these systems would work, and the characteristics of the generated essays, is thus needed in order to detect them. This paper describes the components we believe an automated essay generator would need to have, and the results of building a prototype of the first of these components, the Gatherer.

Keywords: Automated essay writing, Automated essay grading, Gatherer.

1 Motivation for the Study

One of the authors (RW) has had extensive experience in building and testing an automated essay grading system. The other author (JN) thought about the logical extension of this technology to the automatic writing of essays. Subsequently one author visited the other for several weeks in 2008 during which his ideas were tested. This paper describes the background to essay writing and scoring, and the experiences with the prototype Gatherer system, the first component of the proposed automated essay writer.

2 Essay Scoring Technology

Computer based assessment began in 1955 when Lindquist developed optical test-scoring equipment at the University of Iowa. Large-scale testing programs, involving millions of students at all educational levels, are now commonplace. These programs are made efficient and effective through the use of computer and scanning technology [1]. This equipment however is only suitable for True-False and Multiple Choice questions, commonly known as Objective tests. Objective tests can measure many learning outcomes, however

...there remain significant instructional outcomes for which no satisfactory objective measurements have been devised. These include such outcomes as the ability to recall, organize, and integrate ideas; the ability to express oneself in writing; and the ability to supply rather than merely identify interpretations and applications of data. [13].

3 The Value of Essays

According to Ebel essay tests

...provide a better indication of students' real achievements in learning. Students are not given ready-made answers but must have command of an ample store of knowledge that enables them to relate facts and principles, to organize them into a coherent and logical progression, and then to do justice to these ideas in written expression. [9].

Essays also provide an indication of the nature and quality of students' thought processes, as well as their ability to argue in support of their conclusions [10]. The relative merits of Objective tests and Essay tests are summarized by Ebel as follows:

An essay examination is relatively easy to prepare but rather tedious and difficult to score accurately. A good objective examination is relatively tedious and difficult to prepare but comparatively easy to score. [11].

We can conclude then that computer support for scoring objective tests is widely available, but that essay testing may be preferred for measuring the higher level abilities of students. If essays could also be graded by computers, then the time consuming tasks of human grading could be reduced and efficiencies in grading could be obtained similar to that obtained for objective tests. Computer grading of essays is now possible, and the accuracy of the grading can match that of humans. The question then arises as to whether students could obtain software tools to automatically write essays and fool the automated grading systems.

University students have always been required to write essays for assessment. An essay topic, expected length, and due date are generally specified by the lecturer. The student is then expected to research the topic, think about the issue, and write his/her response. The student has to be careful about plagiarism, and to correctly reference

source material. Essays are generally used when the lecturer wants to assess the student's ability to express and synthesize ideas, which cannot be measured by multiple choice or short answer tests.

4 Essays for Sale

Students today have available to them many World Wide Web (Web) sites that can provide an essay for a fee. Sites include

- Custom Writing: <http://custom-writing.org/>
- CustomEssays.co.uk: <http://customessays.co.uk/>
- Prime Essay: <http://www.primeessays.com/>
- Tailored Essays: <http://www.tailoredessays.com/>
- Order Papers.com: <http://www.orderpapers.com/>
- OvernightEssay.com: <http://overnightessay.com/>

These sites provide essays from databases of pre-written essays, or writers will write custom essays to order. Turnaround time can be as little as three hours. Detection of these bought essays is difficult because we assume that they are not published to the Web and hence cannot be detected by search engines.

5 Automatically Grading Essays

Essays can now be graded automatically by specialised software. We know of sixteen different systems, which are listed below.

1. AutoMark [20]
2. Bayesian Essay Test Scoring System [24]
3. Conceptual Rater [5]
4. Content Analyst [8]
5. Educational Testing Service 1 [3]
6. Electronic Essay Rater [4]
7. Intelligent Essay Assessor [15]
8. Intelligent Essay Marking System [19]
9. Intellimetric [27]
10. Blue Wren Software [2]
11. Paperless School Free Text Marking Engine [17]
12. Project Essay Grade [22][23]
13. Rx Net Writer [6]
14. SAGrader [14]
15. Schema Extract Analyse and Report [7]
16. Text Categorisation Technique [16]

These systems make use of natural language processing technology and statistical techniques to analyse style and/or content. Some of the systems typically use between fifty to four hundred human graded essays to train the systems for the specific essay

questions. Multiple linear regression is often used to build a scoring equation from the linguistic and content features of these training essays. Ungraded essays are then assigned a score using this equation, and the specific predictor values for each essay. Most of these systems can perform as well as human markers in the sense that the computer-human score correlations are similar to the human-human score correlations on the same essays. The systems' computer-human score correlations tend to be between 0.70 and 0.90. One of the authors (RW) has developed an essay grading system [2]. One test of the system with several hundred essays of about four hundred words in length achieved a computer-human score correlation of 0.79 compared with the human-human score correlation of 0.81 [28]. These systems are starting to be deployed in primary and secondary schools, as well as universities. For technical details about some of the major commercial systems, and their performances, see [25] [26]. For critical evaluations of some of these systems see [12].

6 Computer Generated Essays

Essay processing technology is now starting to incorporate essay-writing systems. Perhaps the best known system is SCIGen (<http://pdos.csail.mit.edu/scigen/>), a system to randomly generate computer science research papers. A paper generated by the system was accepted as a non-reviewed paper at a conference in 2005. The question then becomes whether automated essay writing systems can generate intelligent and coherent essays which can fool university markers into assigning good grades to them. A second question is whether we can identify characteristics of automatically generated essays, and then flag the essays for the attention of the human graders. In order to understand this problem one of the authors decided to build an automated essay writing system and get a feel for these distinguishing features, if they exist. This system, GhostWriter, is currently under development.

We think an essay writing system should have the following functionality:

- A Gatherer to search the Web for documents relevant for the essay topic, retrieve these documents, and then assign a score for the degree of relevancy.
- The Organizer to select and assemble the appropriate sections of the retrieved documents which will form the body of the essay.
- Templates for defining the essential structure of the essay.
- The Compositor to build the essay from the retrieved material.
- A Spelling and Grammar Checker to standardize the grammar of the essay, and to correct spelling errors.
- A Reviewer tool that allows for quality checking.
- A Distortion module to mask the text copied from the Web documents in order to make the essay unique.

We have so far developed a prototype Gatherer, and in this paper we discuss its architecture, and the results of some testing we have performed with it. The Gatherer has utility outside automated essay writing, so is of value by itself as a form of meta-search tool.

7 Architecture of the Gatherer

The Gatherer takes as its input keywords that relate to the required essay topic. It also needs the user to specify search engine sites which will be used by the Gatherer to find the relevant documents. A simple Web page generated with some PHP scripting is used for obtaining the input controls i.e., the search terms. This information is then passed onto a Perl script that performs all the main required tasks and generates a simple Web page allowing the results to be accessed. This page is being enhanced to permit cleanup of temporary storage and better management of the searches. For example, it would be useful to be able to modify and rerun the search. Currently all keywords or phrases have the same weighting in terms of the search, and the system does not have a built-in Boolean logic ability to fine tune the query.

While the present proof-of-concept has deficiencies as mentioned, it has the particular strength of being compact and easily modified. We intend it to be an open-source tool and will shortly be making it available on the Web. Our approach is to use as far as possible the public face of search engines rather than their particular APIs (Application Program Interfaces). This reduces the risk that the API service will be discontinued (as several have been, e.g., Ask disabled their API in March 2007). However, the Web “face” of the search tools can change, and will force changes to our script.

We have attempted to structure the Perl script as a backbone with plug-in modules for each search tool and each document conversion. We intend to do the same for methods for scoring the retrieved documents, as ultimately we hope to be able to compare scoring strategies. We welcome collaborations.

8 Searching the Web

The following Web resources are currently used for searching:

- Wikipedia in English.
- Yahoo! News.
- Google (We are also considering the Google Scholar service, but are not sure this will remain open.).
- Ask.com.

It is envisioned that in the future other document repositories will be added to the system. For example, we have identified the Social Science Research Network collection of papers (<http://www.ssrn.com/>) and the ACM Digital Library (<http://portal.acm.org/dl.cfm>) as potentially useful resources. The automated script for finding and then retrieving the Web documents uses the Web tool called WGET [21]. This method also saves the documents in files on the system’s computer hard disk.

9 Document Formats

Common document types found on the Web include the following:

- HTML – hypertext mark-up language
- .txt – text file

- .pdf – Portable Document Format file
- .doc – Microsoft Word document
- .odt – Open Office Writer document
- .ppt – Microsoft PowerPoint presentation
- .odp – Open Office presentation

The system being discussed in this paper obviously needs to be able to process these multiple document formats, as a typical search will return a mixture of these file types. The system uses HTML as its base format for all processing. When .txt files are found, they are processed as though they were in HTML format. PDF files are converted using the open-source, cross-platform tool `pdftohtml`. (<http://pdftohtml.sourceforge.net/>). Proprietary formats, such as Microsoft Word and Open Office Writer documents, can be converted to HTML using APIs present within these applications, and/or by third party tools. There may be issues with platform-independence with some of these document types however. For example, Microsoft Word documents cannot be handled directly by Unix/Linux platforms. Currently the system only converts PDF files, but the code is structured in a manner that lets us plug in other converters as needed.

Once the documents have been retrieved and converted to HTML, some editing of the documents takes place. Lowercase letters are converted to uppercase in search terms. Newlines tags are removed, multiple spaces are replaced with single spaces, and some special characters are also removed. Some simple analysis of the documents then takes place. We count the number of times the search word or phrase occurs, and take note of their positions within the document. At a future stage it is intended to make use of this information for scoring the relevance of the document, as our present algorithm is very crude and makes some obvious mistakes.

10 Observations, Ongoing Work and Conclusions

Our work in building a prototype Gatherer, and the testing of it we have undertaken, has indicated to us that the system is quite useful, not only as a component of the proposed GhostWriter, but also for other applications that require Web searches for documents on a particular topic. The Gatherer, in its current form, can be used not only for finding the relevant documents – it also downloads them, saves them, and converts them into HTML. This is particularly useful for researchers, students, and people in industry who wish to prepare research reports on particular topics.

Proposed future development of the system includes a wider selection of Web sites for searching, the ability to convert documents to other formats than HTML, and to improve the platform independence of the Gatherer. We also want to build a better scoring algorithm which will indicate the relevance of the documents for the chosen topic. We also hope to build prototypes for the other components of GhostWriter.

Acknowledgements. The authors acknowledge the support of Curtin University of Technology for partial support for Professor Nash's academic visit in January and February 2008. Professor Nash is also supported by a grant in lieu of salary from the University of Ottawa during his sabbatical. The Telfer School of Management hosts

the macnash server. Both Curtin University of Technology and the University of Ottawa have provided network and physical support allowing this work to proceed. Mary Nash kindly proofread a draft. Discussion and suggestions from Christian Guetl have been very helpful.

References

1. Baker, F.B.: Automation of Test Scoring, Reporting, and Analysis. In: Thorndike, R. (ed.) *Educational Measurement*, 2nd edn., American Council on Education, Washington, D. C (1976)
2. Blue Wren Software Pty. Ltd., <http://trial.essaygrading.com>
3. Burstein, J., Kaplan, R., Wolff, S., Lu, C.: Using Lexical Semantic Techniques to Classify Free-Responses. In: *Proceedings from the SIGLEX 1996 Workshop*. ACL, Santa Cruz (1996)
4. Burstein, J., Kukich, K., Wolff, S., Lu, C., Chodorow, M.: Enriching Automated Essay Scoring Using Discourse Marking. In: *Proceedings of the Workshop on Discourse Relations and Discourse Markers, Annual Meeting of the Association of Computational Linguistics*, Montreal, Canada (August 1998)
5. Burstein, J., Leacock, C., Swartz, R.: Automated Evaluation of Essay and Short Answers. In: Danson, M. (ed.) *Proceedings of the Sixth International Computer Assisted Assessment Conference*, Loughborough University, Loughborough, UK (2001)
6. California Electronic Writer/V.A.F., http://www.rxnetwriter.com/product_sheet.html
7. Christie, J.R.: Automated Essay Marking – For Both Style and Content. In: Danson, M. (ed.) *Proceedings of the Third International Computer Assisted Assessment Conference*, Loughborough University, Leicestershire, UK (1999)
8. Content Analyst Company, <http://www.contentanalyst.com/solutions/essay.htm>
9. Ebel, R.L.: *Essentials of Educational Measurement*, 3rd edn., p. 96. Prentice-Hall, Englewood Cliffs (1979)
10. Ebel, R.L.: *Essentials of Educational Measurement*, 3rd edn. Prentice-Hall, Englewood Cliffs (1979)
11. Ebel, R.L.: *Essentials of Educational Measurement*, 3rd edn., p. 100. Prentice-Hall, Englewood Cliffs (1979)
12. Ericsson, P.F., Haswell, R. (eds.): *Machine Scoring of Student Essays - Truth and Consequences*. Utah State University Press, Logan (2006)
13. Gronlund, N.E., Linn, R.L.: *Measurement and Evaluation in Teaching*, 6th edn., p. 211. Macmillan, New York (1990)
14. Idea Works, <http://www.ideaworks.com/sagrader/>
15. Landauer, T.K., Foltz, P.W., Laham, D.: An Introduction to Latent Semantic Analysis. *Discourse Processes* 25, 259–284 (1998)
16. Larkey, L.S.: Automatic Essay Grading Using Text Categorization Techniques. In: *Proceedings of the Twenty First Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Melbourne, Australia, pp. 90–95 (1998)
17. Mason, O., Grove-Stephenson, I.: Automated Free Text Marking with Paperless School. In: Danson, M. (ed.) *Proceedings of the Sixth International Computer Assisted Assessment Conference*, Loughborough University, Leicestershire, UK (2002)

18. McGee, T.: Taking a Spin on the Intelligent Essay Assessor. In: Ericsson, P.F., Haswell, R. (eds.) *Machine Scoring of Student Essays - Truth and Consequences*. Utah State University Press, Logan (2006)
19. Ming, P.Y., Mikhailov, A.A., Kuan, T.L.: Intelligent Essay Marking System. In: Cheers, C. (ed.) *Learners Together*. NgeeANN Polytechnic, Singapore (2000)
20. Mitchell, T., Russell, T., Broomhead, P., Aldridge, N.: Towards Robust Computerised Marking of Free-Text Responses. In: Danson, M. (ed.) *Proceedings of the Sixth International Computer Assisted Assessment Conference*, Loughborough University, Leicestershire, UK (2002)
21. Nikšić, H., Cowan, M.: GNU Wget, <http://www.gnu.org/software/wget/>
22. Page, E.B.: The Imminence of Grading Essays by Computer. *Phi Delta Kappan*, 238–243 (January 1966)
23. Page, E.B.: Computer Grading of Student Prose, Using Modern Concepts and Software. *Journal of Experimental Education* 62, 127–142 (1994)
24. Rudner, L.M., Liang, T.: Automated Essay Scoring Using Bayes' Theorem. *Journal of Technology, Learning, and Assessment* 1(2) (2002)
25. Shermis, M.D., Burstein, J.C. (eds.): *Automated Essay Scoring - A Cross-Disciplinary Perspective*. Lawrence Erlbaum Associates, Mahwah (2003)
26. Valenti, S., Neri, F., Cucchiarelli, A.: An Overview of Current Research on Automated Essay Grading. *Journal of Information Technology Education* 2, 319–330 (2003)
27. Vantage Learning, <http://www.vantage.com/pdfs/intellimetric.pdf>
28. Williams, R.: The Power of Normalised Word Vectors for Automatically Grading Essays. *Journal of Issues in Informing Science and Information Technology* 3, 721–729 (2006)

Attitudes Toward ICT of Electronic Distance Education (ePJJ) Students at the Institute of Education Development, University Technology Mara

Che Zainab Abdullah, Hashim Ahmad, and Rugayah Hashim

Institute of Education Development
University Technology Mara (UiTM)
40450 Shah Alam, Selangor, Malaysia
Tel.:+(603) 5522-5372/5378/5493
Fax: (603) 55225319
guy73106@yahoo.com

Abstract. The purpose of this study was to assess the attitudes toward information and communication technology (ICT) of adult students undertaking further studies through the medium of electronic distance learning. Attitudes were studied in an attempt to ascertain factors such as anxiety, confidence, liking and, usefulness at the diploma and undergraduate levels.

A total of 500 adult students at various stages of study from diploma to undergraduate degrees participated in this research. The response rate was 56.8%. The instrument used is a replicated questionnaire designed by Loyd and Gresard 1988. Prior permission was granted for the use of the questionnaire which consisted of solicited demographic variables and 40 attitude-based statement. The statements were structured on a scale of one to five. Parametric and non-parametric analyses were executed by using SPSS. The results suggested that the adult students exhibited positive attitudes toward ICT in terms of usefulness and liking, but, semblances of low confidence and anxiety were also evidenced in the statistics.

Keywords: Computer attitude, e-learning, ICT usage, education development.

1 Introduction

1.1 Background

Malaysia looks to education as the key to its socioeconomic development particular in the Knowledge Economy. Furthermore, with a market-sensitive education system, it is Malaysia's strategy to be the education hub of Asia through ICT-enhanced teaching and learning as a common place for interaction [3], [10]. The effective application and exploitation of information technology for national socioeconomic growth and development in Malaysia is now at a critical state. Emerging cultural, social and economic trends arising from the pervasive use of ICT have indicated that information and knowledge of computers are also strategic factors besides land, labor, capital and entrepreneurship in determining the future potentials of a nation

aspiring to be a developed one in the year 2020. The Ministry of Education has responded by implementing wide-ranging reforms to give schools, universities and other higher education institution the skills and competence to ride the crest of the ICT wave. Henceforth, it is crucial that UiTM students who have enrolled in distance education mode, realize the ubiquity of ICT in e-learning. Therefore, the objective of this study is to determine the students' attitudes toward ICT relative to age, qualification, program registered for, gender, and computer skill level.

2 Statement of the Problem

The problem of the study was determining the attitudes toward ICT of distance education learners at the Institute of Education Development (InED), UiTM. This problem cropped up when these students requested to have more face-to-face seminars when their mode of learning and teaching was through the use of ICT. By having more traditional teaching would defeat the purpose of having e-learning for InED, UiTM. In addition, feedbacks received from the distance learners through InED's public forum showed that the use of computers as a mode of education exchange do not augur well for them. Thus, this study was conducted at an appropriate time as both parties need to have a win-win situation. The identification of attitudes relating to age, education background, program registered for, gender, work sector and level of computer usage would support the research hypotheses. Also, the relationship of these demographic variables with computer usefulness, confidence, liking and anxiety would provide sufficient empirical evidence for InED to adjust to the students needs. Furthermore, the findings from this study would be relevant as one of the sources of reference for other institutions of higher learning that offers e-learning programs and courses. By improving the condition of the curricula, the top management of InED and UiTM would be able to ascertain the ICT needs and trends and to suggest recommendations for changes.

3 Methodology

This study will employ the usual traditional approach to descriptive and practical research with quantitative analyses used to derive the empirical evidence that would answer the research questions [1], [2], [4], [8], [7]. Cross sectional and convenience sampling techniques were used to determine the scope and unit of analysis [4], [8], [7]. The instrument was also pre-designed, that is, the questionnaire was replicated from Loyd and Gressard's [6] and Hashim et al [5] and amended to suit the current environment. For this study, the Cronbach Alpha score was 0.869, which means that this questionnaire is valid and reliable.

4 Findings

The response rate to the 500 questionnaires administered was 56.8 % or 284 useful data. The minimum, maximum, mean and standard deviation of these four variables are shown in Table 1 where anxiety showed the highest mean score of 28.87, followed by liking at 28.02. The variable, 'confidence' came in third at 27.62 and last was usefulness at 25.98. The graphical presentation of these subscores are shown in Figure 1

Table 1. Descriptive Statistics for Attitudes' Subscores

	Min	Max	Mean	Std. Deviation
Anxiety	20.00	99.00	28.8718	15.54801
Confidence	19.00	99.00	27.6207	15.84088
Liking	15.00	99.00	28.0199	16.22616
Usefulness	14.00	99.00	25.9829	15.07485

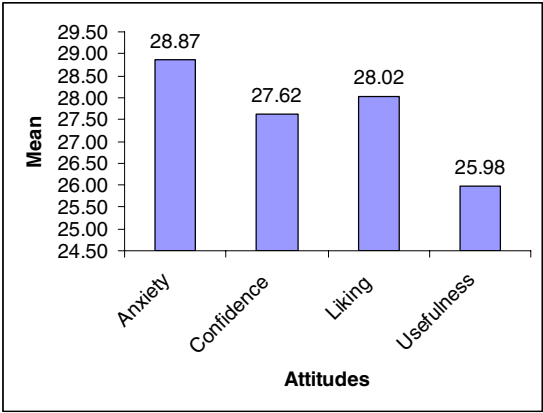


Fig. 1. Bar Chart for Mean Subscores of the four Attitudes

Furthermore, a t-test was conducted to compare the means of the four attitude variables or subscores as shown in Table 2. The results indicated that there were significant differences between the four means, that is, the significant value (*p*-value) is less than 0 ($p < 0$). Also, the mean scores for the four attitudes are: usefulness = 25.98, confidence = 27.62, liking = 28.02 and anxiety = 28.87. In the questionnaire, the attitudes are measured on a scale of 1 to 5 (Strongly Disagree to Strongly Agree) (Table 2).

- a) Next, from the output in Table 3, it is found that there is no difference between gender and attitudes toward ICT with respect to computer anxiety, confidence, liking, and usefulness. All four attitudes have Pearson Chi-square significant values well above the alpha level of 0.05 where usefulness = 0.557, confidence=0.120, liking=0.094 and anxiety=0.686. Therefore, the null hypothesis is accepted as all the *p*-values are greater than 0.05; and all the alternative hypotheses are rejected. Furthermore, the minimum expected cell frequencies for usefulness = 28, confidence = 27, liking = 28 and anxiety = 28 which are all greater than five (>5). Thus, we can be confident that we have not violated one of the main assumptions of chi-square.

Therefore, in examining the observed cell frequencies, it can be concluded that gender do not show a significant difference for any of the four attitudes, $X^2(16, N=339) = 14.558, p>.05$; $X^2(13, N=336) = 19.096, p>.05$; $X^2(16, N=339) = 23.787, p>.05$; and $X^2(15, N=339) = 11.909, p>.05$ respectively.

Table 2. Comparison of Means (t-test) for the four Attitudes

	Test Value = 0					
	T	Df	Sig. 2-tailed	Mean Diff.	95% Confidence Interval of the Difference	
					Lwr	Uppr
Anxiety	34.8	350	.000	28.9	27.2	30.5
Confidence	32.5	347	.000	27.6	26.0	29.3
Liking	32.3	350	.000	28.0	26.3	29.7
Usefulness	32.3	350	.000	26.0	24.4	27.6

Table 3. Chi-Square Tests for Gender & Attitudes

Chi-Square Tests: Gender * Usefulness

	Value	Df	Asymp. Sig. (2-sided)
Pearson Chi-Square	14.558(a)	16	.557
Likelihood Ratio	16.472	16	.421
Linear-by-Linear Association	.059	1	.808
N of Valid Cases	339		

a 18 cells (52.9%) have expected count less than 5. The minimum expected count is .28.

Chi-Square Tests: Gender * Confidence

	Value	Df	Asymp. Sig. (2-sided)
Pearson Chi-Square	19.096(a)	13	.120
Likelihood Ratio	22.181	13	.053
Linear-by-Linear Association	.005	1	.942
N of Valid Cases	336		

a 11 cells (39.3%) have expected count less than 5. The minimum expected count is .27.

Table 3. (Continued)

Chi-Square Tests: Gender * Liking

	Value	Df	Asymp. Sig. (2-sided)
Pearson Chi-Square	23.787(a)	16	.094
Likelihood Ratio	25.410	16	.063
Linear-by-Linear Association	3.008	1	.083
N of Valid Cases	339		

a 17 cells (50.0%) have expected count less than 5. The minimum expected count is .28.

Chi-Square Tests: Gender * Anxiety

	Value	Df	Asymp. Sig. (2-sided)
Pearson Chi-Square	11.909(a)	15	.686
Likelihood Ratio	12.730	15	.623
Linear-by-Linear Association	.471	1	.492
N of Valid Cases	339		

a 16 cells (50.0%) have expected count less than 5. The minimum expected count is .28.

5 Discussions

The survey results indicated that anxiety has the highest mean score of 28.87, followed by liking, confidence and usefulness. Therefore, it can be safely assumed that the ePJJ students were apprehensive and probably ‘technophobic’ toward ICTs. Hence, the Institute of Education Development (InED), UiTM should look into this matter seriously because non-usage of computers in distance education or e-learning defeat the purpose offering flexible learning programs and investing in ICT. Close to the heel of anxiety is the attitude, liking. This indicated a high positive attitude, which meant that the ePJJ students like using ICTs but were anxious and unsure of what to do with certain features in customized software particularly InED’s learning management systems (LMS). InEd’s current LMS is called *i-class*. Proper training should overcome this attitude. But, on average, confidence and usefulness show high mean scores, that is, the students have positive attitudes toward ICT. Henceforth, the implications from the above discussion and of researching on attitudes toward ICT would involve long term benefits and strategic exploitation of ICT investment and the future of e-learning. It is important to remember that ICT is a tool or an enabler towards better delivery of education,

but the user is the key. If the students exhibit negative attitudes toward ICT, then e-learning would not be their choice of seeking higher education.

6 Conclusion

To conclude, ICT is the foundation for e-learning. Without ICT there would obviously be no e-learning. In distance education, ICT is the enabler for most means in imparting education. Hence, the requests by students to have more face-to-face seminars rather than online teaching should not be catered to. Furthermore, the findings from this research proved that attitudes toward ICT is more of the selfish nature of the adult students'. If traditional teaching is preferred, then being a full-time student would be the solution.

7 Recommendation

It is recommended that a needs assessment be conducted in order to determine the various components of computer training for the students since the results indicated high anxiety when using ICT and working with computers among the Also, the findings from the survey indicated that students with no experience in ICT usage have more negative attitude towards ICT. Further research should be conducted to determine the reasons for this and to suggest possible solutions. Thus in line with Ward and Peppard [13], the following are suggested:

1. Perceived credibility gap between the 'hype' of the ICT industry and what ICT can actually do and how easy it is to do it. Given these difficulties, InED may not be able to claim the benefits offered by ICT.
2. Despite the difficulty in expressing all ICT benefits in economic terms, InED, UiTM and the Ministry of Higher Education should not demand to see financial justification for investments in ICT. Producing quality graduates who are skilled in most aspects of ICT usage should be the objective of InED. Producing better workforce to meet Malaysia's market needs would also ensure other economic gains from other stakeholders. Ensuring better ICT infrastructure and Internet access would cushion the impact of globalization, yet generate wealth through knowledge and information.

References

1. Beins, B.C.: *Research Methods: A Tool for Life*. Pearson, Boston (2004)
2. Blaikie, N.: *Analyzing Quantitative Data: From Description to Explanation*. Sage, Thousand Oaks (2003)
3. Capron, H.L.: *Computers: Tools for the Information Age*. Benjamin-Cummings, Menlo Park (1987)
4. Coakes, S.J.: *SPSS: Analysis Without Anguish: Version 12.0 for Windows*. Wiley, Queensland (2005)
5. Hashim, R., Rahman, A.L.: *A and Kassim, A, Antecedents of Computer Attitudes: A Case of the Royal Malaysia Police, Sabah*. Unpublished research paper (2007)

6. Loyd, B.H., Gressard, C.P.: The nature and correlates of computer anxiety in college students. *Journal of Human Behavior and Learning* 3, 28–33 (1988)
7. Heiman, G.W.: *Understanding Research Methods and Statistics: An Integrated Introduction for Psychology*, 2nd edn. Houghton Mifflin, Boston (2001)
8. Sekaran, U.: *Research Methods for Business: A Skill-Building Approach*, 4th edn. John Wiley & Sons, Inc., Singapore (2003)
9. Stangor, C.: *Research Methods for the Behavioral Sciences*, 2nd edn. Houghton Mifflin, New York (2004)
10. Ward, J., Peppard, J.: *Strategic Planning for Information Systems*, 3rd edn. Wiley, England (2002)

Metaverse Services: Extensible Learning with Mediated Teleporting into 3D Environments

Ioakim Marmaridis and Sharon Griffith

University of Western Sydney, Sydney, Australia

Abstract. Metaverse-enabled Learning Spaces (MeLS) offer significant improvements over traditional modes of teaching allowing for experiential learning. On the other hand, there is large number of capabilities the immersive 3D world of Metaverse-enabled spaces has to offer to traditional web-based tools for learning. Unfortunately, those capabilities are not being leveraged. We build upon a number of open source learning packages and offer an architecture for Metaverse services extending learning capabilities by offering 3D environments as services inside and in combination with more traditional web-based systems. The goal is to allow educators and students to easily move from traditional web-based systems into MeLS experiencing environments, which are not available in a traditional LMS software package, but can be offered or enhanced by the Metaverse presentation layer. This is made possible by our method for performing mediated teleporting into the MeLS.

1 Introduction

Need for enriched education media coupled with advances in technology have given rise to Metaverse-enabled learning spaces (MeLS). Users typically access the online system with a proprietary client and interact with in-world avatars and metaverse elements [1]. This virtual platform offers connectivity with other Internet resources, such as web pages, RSS feeds, wikis and blogs.

Metaverse-enabled Learning Spaces (MeLS) offer an immersive environment and allow for the simulation of otherwise very expensive or simply impractical items such as buildings, bridges or works of art [2,3]. MeLS represent most of their structures as 3D rendered objects and also allow students to manipulate and build on top of existing objects. The modification tools offered in the MeLS include simple object construction and scripting utilities, which enable users to design and create 3D primitives with interactivity [4].

Work is under way to integrate functionality from existing e-learning software packages - such as Moodle[5], WebCT, Blackboard and others - with MeLS. In this paper, we discuss the possibility of using MeLS where appropriate to enhance and augment the learning experience of students through a rich interactive online environment.

To achieve this goal, we propose an architecture for linking from within existing e-learning tools such as a LMS and virtual classroom tools directly into MeLS. We showcase our additions to existing technologies such as SLURLs [6] in

order to track access and provide the infrastructure necessary to further control the student experience and offer support tools for them while transitioning from the mainstream e-learning environment into the MeLS.

Our proposed architecture also addresses the need for educators to provide additional structure for in-world activities students can undertake [7]. To that end, we offer facilities for tracking student access from the traditional tools to MeLS while offering suggestions for how this new data dimension can be used by educators in creative ways.

2 Background and Motivation

The authors strongly believe in the potential MeLS offer for educators to enrich the learning experience of their students; particularly those who are used to accessing learning via the Internet [8]. We found however, that existing technologies, although capable of carrying out the core functionality, they lack in providing the refined control and management for the educators whilst they setup the interactive environment for their students.

Before we describe the details of our proposed additions, we would like to offer some definitions that are commonly used along with some explanation of their origin.

The Metaverse is a virtual world, described in Neal Stephenson's 1992 science fiction novel *Snow Crash*, where humans, as avatars, interact with each other and software agents, in a three-dimensional space that uses the metaphor of the real world. The word Metaverse is a compound of the words "meta" and "universe". Users of the Metaverse can experience the environment from both a first person or third person perspective.

Learning management systems (LMS) are a category of software systems capable of delivering training materials via the World Wide Web (web). The degree of complexity within the LMS can vary according to the complexity of the on-line learning content served. Most LMS are web-based, and can be accessed via a web browser allowing their users to gain ultimately access to online learning content. Content may be professionally prepared and purchased or it can be user generated or a combination of both. One of the popular features of a LMS is its ability to offer templates and simple forms to create interactive web-based class environments [9]. There is a large variety of LMS applications available, ranging from multi-million dollar enterprise-level ones down to free and open source ones. Most academic institutions (if not all) use some LMS system and so do most medium and large businesses to manage training of their staff. With Metaverses steadily increasing in popularity as learning platforms, the need for better integration between these two categories of software is also increasing.

Our motivation for this work is to provide an overview of the recent approaches to integrate 3D and 2D learning spaces that provide users the rich social interaction of Metaverses with a traditional learning management system [10]. We also propose a facility for tracking, verification and validation when integrating users between the two learning spaces.

3 Existing Approaches and Considerations

Work is already under way to offer various forms of integration between LMS and Metaverse systems, one of the most prominent projects is called Sloodle [11] and it is an attempt to integrate the Moodle LMS with the Second Life (SL) Metaverse. The integration takes the form of custom additions to Moodle and a number of purpose-built objects within SL that are capable of communicating with each other at near real time.

3.1 Sloodle

Sloodle focuses heavily on creating special objects in-world that attract students and allow them to indirectly interface with the LMS backend and some of its built-in functionality. In this paper, we aim at streamlining and enhancing the movement of students the other way - from the e-learning tools to MeLS.

Figure 1 shows a view of Sloodle taken from within Second Life.



Fig. 1. Sloodle in-world environment

Even though Sloodle offers a set of objects that are used in-world to make traditional LMS services accessible to students, it requires that students find their own way into the places within SL where those objects are located and where they can interact with them.

On the other hand, SL is quite an extensive place and therefore students need to be guided as to where within the Metaverse they need to visit in order to observe various objects or study parts of the virtual world as part of their learning experience. There is therefore the need for a facility allowing students to visit specific parts of SL while they happen to be accessing the LMS in a

traditional way and this access must be controlled and tracked by the educator in charge of delivering the learning to the students.

3.2 No Integration

The simplest perhaps approach to the above issue, is to simply give to students the co-ordinates of the location in the virtual world and ask them to find their way there. This would involve starting up their client for SL, authenticate themselves and then have to teleport and/or search for the particular location.

Several disadvantages exist within this approach which include:

- Students getting lost / not attending the location within the MeLS
- Students accessing the MeLS and subsequently being side-tracked by other places therein [7]
- Misinterpreting the coordinates or otherwise transporting themselves to the wrong place which may or may not be appropriate for them to access / engage with [12]
- No tracking of the student's activities is possible, the educator cannot know who got to the designated location, when and, in what order.

3.3 Background to SLURLs

Another, slightly better method than manually passing the co-ordinates to students, is through the use of a SLURL (Second Life URL) [6]. SLURLs offer the core method for transferring a user into SL via a specially crafted hyperlink that interacts with the SL client installed onto the users' machine. A typical SLURL looks like this:

`http://slurl.com/secondlife/Ahem/50/30/10/`

There are several components that make up a SLURL as follows:

- URL stem - this is predefined and always points to the slurl.com domain and SL
- region - this specifies the region within SL where the user will be teleported to
- x-coordinate - specifies the x coordinate where the user will be teleported to
- y-coordinate - specifies the y coordinate where the user will be teleported to
- z-coordinate - specifies the z coordinate where the user will be teleported to

The Z co-ordinate is optional and can be omitted from a SLURL. In addition to the above, there are several other details that can be specified via the SLURL such as the image and descriptive text that comes up when a users clicks on the SLURL.

When users click on the SLURL in their browser they are taken to a map view of SL with a button that allows them to teleport to. The teleport button behind the scenes passes a URL call to the user's SL client application that will do the actual teleporting of the user to the specified location.

There is a number of methods to construct SLURLs, ranging from manually concatenating parameters to the URL, to auto-creating via software as the case is with our contribution shown in this paper. Finally, there is a SLURL builder that is accessible online and that allows users to create their own SLURLs by providing values for the various parameters available.

The concept of SLURLs provides for a powerful abstraction and a very convenient mechanism for addressing different parts of SL. Unfortunately however it does not provide control, tracking or management features necessary for educators to enhance the experience of their students. In addition, there is no framework provided by SLURLs for assisting students once they reach the SL environment.

Our Service Broker solves these issues by leveraging existing SLURL functionality while augmenting it with additional features that educators can take advantage off right away.

4 Proposed Approach

4.1 Architecture

In our proposed approach we recognise the disadvantages of passing the coordinates via out-of-band means and have chosen to build on the functionality offered by SLURLs. Our contribution consists of an architecture that incorporates a Service Broker for teleporting students to the desired (by the educator) location in Second Life. At the architectural level, the Service Broker is completely agnostic of the implementation details of the SLURL. As a result it can be easily extended to cater for access to other MeLS (such as OpenSimulator for instance), even though our concrete implementation example uses SLURLs and Second Life. Figure 2 shows the interaction of our Service Broker within the context of Second Life and student interactions.

As one can see, the core task of the Service Broker for teleporting is to inspect the request and enhance it according to rules. In our current implementation, these rules constrain the locations that can be accessed, therefore stopping students from going to the wrong place in the MeLS. The Service Broker also simplifies the overall URL instead of seeing a complete SLURL reducing typing errors or errors in transcription, line splitting in emails and other similar issues caused by malformed URLs.

Tracking is another integral feature of the Service Broker. It serves each URL request while it records information about who accessed what location and when. This information allows educators to gain insights such as what students attended the specified location. With this information they are then able to structure assessments around the time dimension where for instance students are lead to access the MeLS and are ranked according to their order of arrival in the designated place, or if they have made it there in the first place or not.

An important features of our architecture is that we allow the educator to setup and maintain the rules relevant to locations they setup and make available to their students. In this manner, the educator has complete control over what

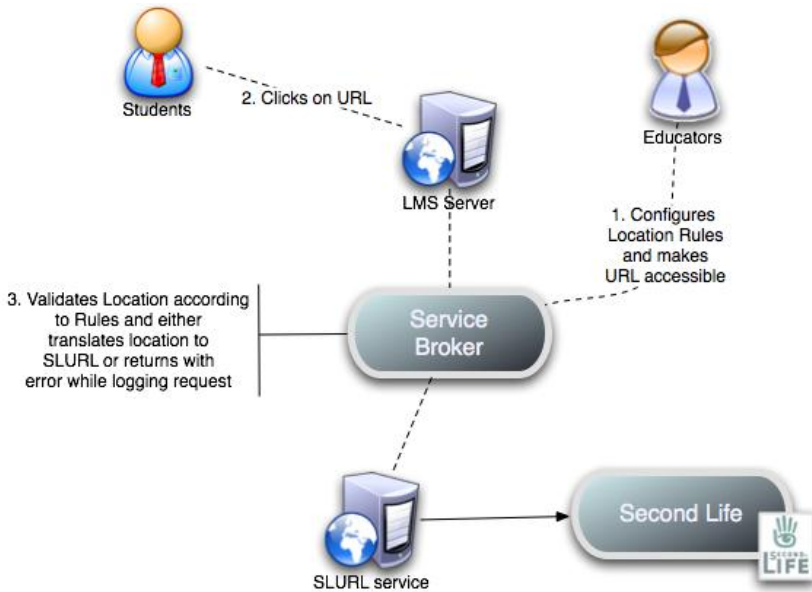


Fig. 2. Service Broker - conceptual view of interactions between students and Second Life

regions the students can access through the URL to the Service Broker she provides them with. In addition to regions, the educator can setup rules around the range of co-ordinates that can be used for x, y, and z.

Finally, rules can be setup that govern how the Service Broker processes a request and what the resulting SLURL will be. This allows educators the freedom to change the logo and introduce custom messages to each location URL they produce, without adding this information into the URL itself, therefore keeping it simple and tamper-proof.

4.2 Implementation

We have selected to build our proof of concept Service Broker using the Perl programming language as it allows for flexibility and rapid development for the WWW. The persistent data of the Service Broker, such as rules and the tracking data are stored in a database back-end powered by MySQL. The architecture is agnostic of technologies and any set of technologies suitable for working with the web can be used to implement this system. Figure 3 shows the lifecycle of a request from a student for teleportation into Second Live.

The lifecycle of each request processed by the Service Broker goes through the same set of steps where it is parsed, validated, and checked against the set of rules that apply to it. If all is successful so far, the Service Broker will log the request along with a date/time stamp and details around who the requester was (based on details retrieved from their web browser / environment variables they

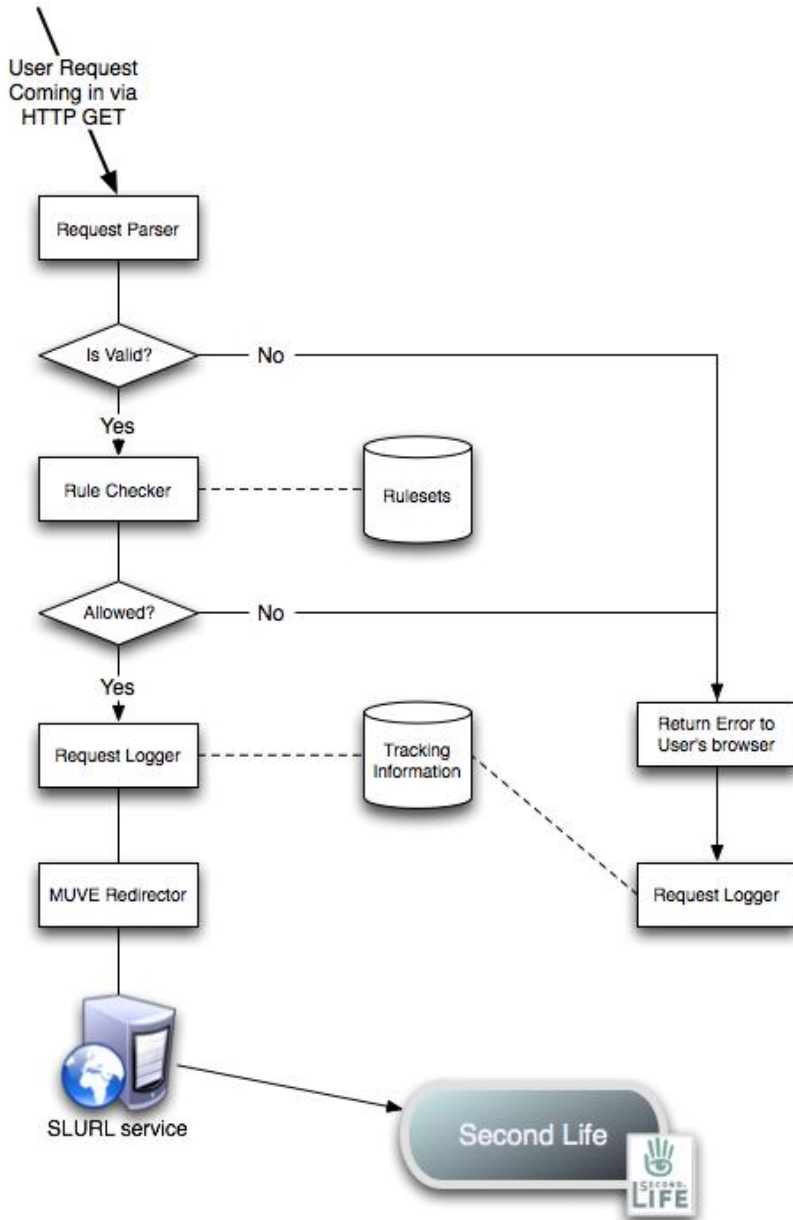


Fig. 3. Service Broker - overview of lifecycle for request for teleportation into Second Life

have set). Finally, the Service Broker will convert the request into a well-formed SLURL and redirect the requesters' browser to it.

From this point onwards the student is dealing with the SLURL as per normal. Given the parameters the Service Broker put into the auto-generated SLURL, the student will see a teleport link similar to the one shown in figure 4.



Fig. 4. Service Broker - the generated teleport SLURL is shown to students before they can enter the Metaverse

Upon clicking on the teleport button, the SLURL will operate as per normal and direct the SL client to transport the student's avatar into the designated location in SL.

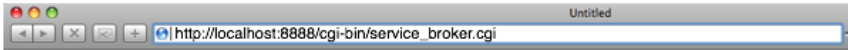
For requests that are unsuccessful, because they are malformed, invalid, or fail according to a rule there is a different execution path inside the Service Broker. These requests will be logged and can be reported on in addition to presenting the student with an error page as shown in figure 5.

At present, this same error message is also shown to students that experience a system error with the Service Broker. There is however provision in the implementation for different errors to be raised and shown for different types of issues.

The tracking provided by the Service Broker is complemented by the reporting functionality built into the system as well. Figure 6 shows a simple report that authorized users can bring up to see what requests the Service Broker has fulfilled successfully and otherwise.

This is quite a simple report showing a number of fields, some of which are auto-generated by the Service Broker at the time of the request processing and some are simply extracted from the request itself and the browser's environment. Specifically the fields shown are as follows:

- user id - taken from the environment variables setup by the application the user was in while clicked the link to the Service Broker
- area requested - this maps to the region in a SLURL and identifies the broader location where the user wished to be teleported

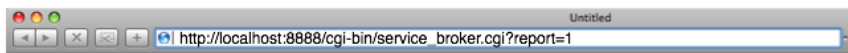


Error - The Service Broker was unable to service your request.

This may be caused by a request for an invalid or inaccessible location or by a system error. Please ensure that you have the correct location URL and try accessing it again in a little while.

If the problem persists, please see your instructor and notify them of this error.

Fig. 5. Service Broker - error message presented to students when their request cannot be fulfilled due to a rule violation



Service Broker - Tracking Report for location requests

User ID	Area Requested	X co-ordinates	Y co-ordinates	Z co-ordinates	Date/Time	Result
student 1	Westopia	50	22		20081024 T1143	ok
student 2	West-op-ia	20	20		20081102 T2208	error - location not supported
makis	SLAC	67	93	10	20081104 T0814	ok
sharon	Westopia	188	45		20081107 T0039	error - location not supported

Fig. 6. Service Broker - view of the built-in reporting mechanism offered

- co-coordinates for x, y, and z - they identify the location itself. Note that the z value is empty for some requests given that this parameter is optional
- date and time stamp - this is auto-generated by the Service Broker as at the time the request was processed
- result - this shows whether the request was successful or not as all requests get tracked by default, including invalid or unsuccessful ones

Our plan moving forward with the development of the Service Broker includes the ability for users to run custom queries via the web interface and produce their own parameter-driven reports. Also we would like to allow for column sorting and a data export facility so users are free to manipulate the data in external software such as a spreadsheet application.

4.3 Benefits

While leveraging existing techniques and infrastructure (found in SLURLs) our addition of mediating the transfer of users through the Service Broker presents a

number of unique advantages for both students and educators. Students are able to move quickly between existing web-based learning tools and SL with URLs that are small and with very few parameters. The URL is short and therefore less susceptible to being mis-read or wrongly transcribed.

Educators, are empowered through the rule-setting capability of the Service Broker and they are now able to construct location specific URLs that they can give their students and be guaranteed that they will have visibility into when their students visit these locations. They are able to view in near real time data around who made use of their link and teleported into SL. Also incorrect attempts are made visible as well and give the educator the ability to proactively approach students who may find themselves in a difficult place accessing the SL location and individually coach them.

Based on those major abilities the Service Broker provides, a number of detailed benefits with a flow-on effect as follows:

- Enriching student experience by making access quicker and less prone to error
- A technology now exists providing robust and transparent access from traditional web-based LMS environments and SL
- Control is placed into the hands of educators giving them data they can mine to uncover patterns in access to each location
- Allowing for an end-to-end mechanism for teleporting into MeLS that can be used by other applications outside the realm of education

5 Conclusion and Future Work

We see the work on Service Broker presented in this paper as only part of the journey towards a web-based environment for mediated access to SL and other MeLS for education and wider uses. To that end, we are exploring a number of items for research and development. For instance, we are looking at methods for providing support for school or institution-wide rules in the Service Broker so that images and descriptions shown to users along with their teleporting link can be enforced at an organisation-wide level rather than by individual educators. Open questions also exist around the volume and detail of data the Service Broker needs to capture. It is possible for instance to have specially made prims (3d objects) in Second Life where they can report the whereabouts of each avatar within the virtual world and have those details recorded within the Service Broker. This would allow for much richer interactions to be designed by the educator where he could design specific activities around students accessing a given area of SL and within it carry out specific activities. Finally, there is investigation necessary in setting up the equivalent of telehubs in existing web applications where for instance Second Life locations used in the context of a course can be aggregated and presented in a page offering an easy overview of the places one would have to visit to successfully complete the course.

Although a number of questions are open and a lot of interesting topics exist for exploration in this area, we believe that the Service Broker presented in

this paper opens up a world of possibilities for leveraging Second Life and other MeLS as educational tools by providing a framework for easy, trackable access for students.

We also see a number of uses of the same concept of mediated teleporting into Second Life from other web applications and systems accessible via the Internet and we are excited by the possibilities that exist in those areas.

References

1. Livingstone, D., Kemp, J.: Integrating web-based and 3d learning environments: Second life meets moodle. *Next Generation Technology-Enhanced Learning*, 8
2. Ye, E., Liu, C., Polack-Wahl, J.A.: Enhancing software engineering education using teaching aids in 3-d online virtual worlds. In: *FIE 2007. 37th annual frontiers in education conference-global engineering: knowledge without borders, opportunities without passports*, 2007, pp. T1E-8 (2007)
3. Leidl, M., Rling, G.: How will future learning work in the third dimension? In: *proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education*, p. 329 (2007)
4. Urban, R.: A second life for your museum: 3d multi-user virtual environments and museums. Status: published or submitted for publication, a *Second Life for Your Museum: 3D Multi-User Virtual Environments and Museums*
5. Wikipedia, Moodle - wikipedia (November 2008),
<http://en.wikipedia.org/wiki/Moodle>
6. LindenResearchInc., Location-based linking in second life (November 2008),
<http://slurl.com/about.php>
7. Kemp, J.W., Haycock, K.: Immersive learning environments in parallel universes: Learning through second life. *School Libraries Worldwide* 14, 89–97 (2008); *immersive Learning Environments in Parallel Universes: Learning through Second Life*
8. Robben, J.J.: Learning environments for the net-generation learners
9. Kemp, J., Livingstone, D.: Putting a Second Life Metaverse skin on learning management systems. In: *Second Life Education Workshop at the Second Life Community Convention, San Francisco, August 20* (2006)
10. Leidl, M., Russling, G.: How will future learning work in the third dimension? *SIGCSE Bull.* 39(3), 329 (2007)
11. Park, H., Jung, J., Sanchez, J., Cha, J., Kim, J., Kim, H.: *Second life for education* (2008)
12. Kluge, S., Riley, L.: Teaching in Virtual Worlds: Opportunities and Challenges. *Issues in Informing Science and Information Technology* 5 (2008)

Combining Simulation and Animation of Queueing Scenarios in a Flash-Based Discrete Event Simulator

Ruzelan Khalid, Wolfgang Kreutzer, and Tim Bell

Department of Computer Science and Software Engineering
University of Canterbury, Private Bag 4800, Christchurch 8140, New Zealand
`rk34@student.canterbury.ac.nz`
{`wolfgang.kreutzer,tim.bell`}@canterbury.ac.nz

Abstract. eLearning is an effective medium for delivering knowledge and skills to scattered learners. In spite of improvements in electronic delivery technologies, eLearning is still a long way away from offering anything close to efficient and effective learning environments. Much of the literature supports eLearning materials which embed simulation will improve eLearning experience and promise many benefits for both teachers and learners. This paper presents an application framework for Discrete Event Simulation (DES), written in Adobe Flash and its *ActionScript 2* scripting language. Based on the framework, a set of DES components has been designed and constructed to ease both construction and embedding of multimedia enriched and animated DES models into web-based eLearning environments.

Keywords: Simulation, animation, Flash based simulator, ActionScript, eLearning.

1 Introduction

Simulation is a technique for experimenting with models of real or imaginary systems [1] and [2]. Since it allows users to manipulate parameters and directly observe the impact of modifications on model behaviour and performance, it can be a powerful learning tool, whose “hands-on” activities engage learners emotionally and help to improve understanding of complex scenarios. There is a large body of literature (e.g. [3], [4] and [5]) that corroborates these benefits of simulation in a teaching and learning environment.

The paper focuses on the design of *tools* to construct *animated* and *interactive* simulation models for web-based delivery. The hoped-for result will be a set of components that help teachers build such models more easily. To achieve this objective, some libraries and a visual environment for composing simulation models have been designed and constructed. Because of its strength as an animation tool [6], [7] and [8] and because it can generate very compact *.swf* applets that can be played “off the shelf” in the vast majority of modern browsers, Adobe *Flash* [9], [10] and [11] was chosen as the implementation tool.

In the project’s initial phase we have concentrated on *Discrete Event Simulation (DES)*, where the state of a model changes only at specified points in time, and on

Queueing Networks, which explore the effects of capacity constrained resources on common performance measures, such as response time and throughput. In this context we primarily investigate tools that foster “modelling for insight” (i.e. models that improve understanding through observation) rather than making quantitative performance predictions (i.e. models that measure how *efficiently* a system performs its functions). In an eLearning context such models can be instructive, since they allow users to visually experiment with changes of model parameters and observe their effects on model behaviour.

Both simulations and animations reflect *change* in either the time or space dimension. *Temporal* change, for example, occurs whenever a simulation encounters delays (in model time) and whenever an animated object changes appearance. *Spatial* change occurs whenever a visual entity *moves*. To support *animated simulations* requires a nested design, where model time must be mapped onto animation time, and animation time must be mapped onto real time. A number of alternative strategies for connecting such layers of representation exist, and we have opted for *synchronous* approach, where model time is always *proportional* to animation time and animation time is always *proportional* to real time. Users have complete control over the speed of both simulation (i.e. the mapping of model time to animation time) and animation (i.e. the mapping of animation time to real time). This allows them to quickly scroll through uninteresting model behaviours and focus closely on interesting events, which may be observed in slow motion. This capability is particularly helpful in a teaching and learning environment.

To coordinate all dynamic aspects of a discrete event simulation, such as time management and event scheduling, a discrete event *monitor* is required and common functionality such as statistical instrumentation, sampling from distributions and presentation of simulation results must be provided. Although there are a number of Java-based simulators e.g. JavaSim [12], Psim-J [13] and Desmo-J [14], and some simple device modelling tools that use Flash (e.g. [15]), we have not found any reports or references to Flash-based discrete event modelling tools. For this reason we have coded our own Flash-based DES model executive, using Flash’ object-oriented scripting language (ActionScript-2, see [16]).

The paper is organized into six chapters. Chapter 2 presents the DES framework and its key packages. In chapter 3 we investigate what graphical objects can best support DES model animation, followed by a brief discussion of Flash components in chapter 4. Finally, to show how teachers can use these simulation components and libraries to construct simulation models, we present a sample implementation of an M/M/1 queueing scenario. Chapter 6 draws some conclusions.

2 The DES Framework

Our framework for modelling queueing scenarios in Flash is derived from the *DEMOS* class library [17] and the SIMFONE [18] modelling framework. Based on their functionality, the resulting Flash components and Actionscript libraries are divided into four packages, i.e. (1) Distributions, (2) Data Collectors, (3) Simulation Executive, and (4) Queues and Resources.

2.1 The Distribution Package

Samples from probability distributions are often used to simulate stochastic model behaviours. The *distributions* package provides a selection of pre-packaged distribution objects. These may, for example, be used to determine the time between workload items’ arrivals or service times of resources. Note that the term “RNG”, used in Figure 1, stands for random number generator. We use Actionscript’s generator, which is based on a standard congruential method, for this purpose.

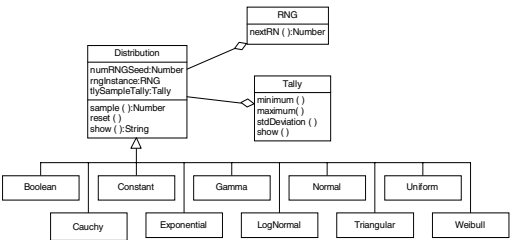


Fig. 1. Class Diagram for the *Distribution* Package

2.2 The DataCollectors Package

Data collectors are used to gather, analyze and report statistical information during a simulation run. The relevant package consists of five classes: *Collector*, *Tally*, *Counter*, *Histogram*, *TimePlot* and *TimePlotEntry*. The *Collector* class forms the base of the data collector hierarchy. *Counters* record relevant changes in model states; e.g. occurrences of significant events. They can, for example, be used to keep track of the number of entities that have entered or left a model. A *Tally* reports the minimum, maximum, mean and standard deviation of a series of values. It can, for example, be used to gather reports on delays; e.g. time spent waiting in queues or residence times in the model. *Histograms* assign values to intervals and show frequency counts for each interval in graphical form (bar charts). *TimePlots* (chronological graphs) are used to track the temporal evolution of a variable’s values; i.e. how they change over time. Plotting the number of entities in a queue or showing changes to a resource’s

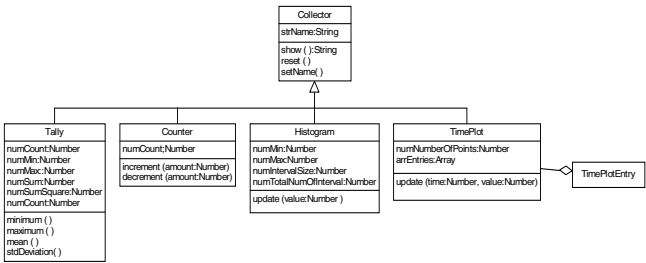


Fig. 2. Class Diagram for the *DataCollectors* Package

utilization during some model time interval can serve as examples. The corresponding Actionscript class uses instances of class *TimePlotEntry* as data points. Figure 2 shows a class diagram for the Data Collectors package.

2.3 The Monitor Package

The *Monitor* package provides the infrastructure for sequencing state transitions in discrete event simulation models. It consists of five classes: *SimProcess*, *Monitor*, *Agenda*, *Clock*, and *Event*. The *SimProcess* class describes the life cycles (i.e. the sequence of events such an entity moves through) of active entities. Since Actionscript 2 does not offer any features for implementing coroutines or threads, each *SimProcess* instance needs to keep track of its current *phase* (i.e. the current stage of its lifecycle). This property is updated whenever the process encounters a model time delay and passes control to the monitor (e.g. see line 30 in Listing 1). The monitor owns an *Agenda* (event list) that maintains a time-ordered list of future events. Whenever new events are scheduled it will insert a process and time reference (event notice) at the appropriate agenda position and wake and remove processes from the agenda whenever their time of occurrence is reached. Instances of the *Event* class are used as agenda entries and store a process reference and a wake-up time (for that process). An awakened process' *phase* value ensures that the process' execution continues from just after the point at which it incurred a delay. A simulation's temporal progress is controlled by the *Monitor* class' single instance, which owns all model components and whose functionality selects the next imminent event from an agenda, updates the model clock (an instance of class *Clock*) to the relevant time value, and activates the appropriate process, instructing it to execute its next phase. Figure 3 shows a class diagram for the Monitor package.

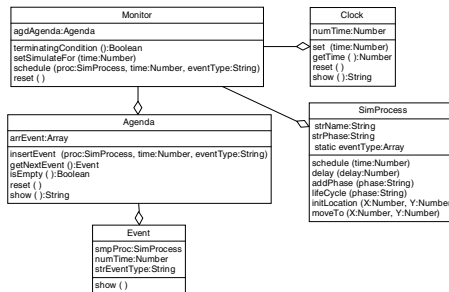


Fig. 3. Class Diagram for the *Monitor* Package

2.4 The Resource Package

Currently this package consists of two classes: *Server* and *Queue*. *Server* allocates limited capacity resources to service requests. If a server's capacity is exhausted, the requesting entity will be placed in a *service queue* - an instance of the *Queue* class. Figure 4 shows a class diagram for the Resource package.

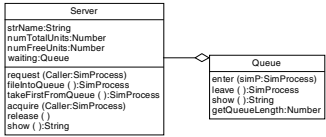


Fig. 4. Class Diagram for the *Resource* Package

3 Graphical Objects in DES Models

Graphical objects for animating a simulation model in a Flash environment can be split into two different categories. The first one is independent of a simulation domain, while the second is specific to a particular type of simulation (see Figure 5). One of the benefits of using Flash as a base is that graphical interfaces to various model functionality can be constructed quite easily. Flash’ drawing tools can also enhance model backgrounds, and suitable text areas and labels can improve a model’s documentation.

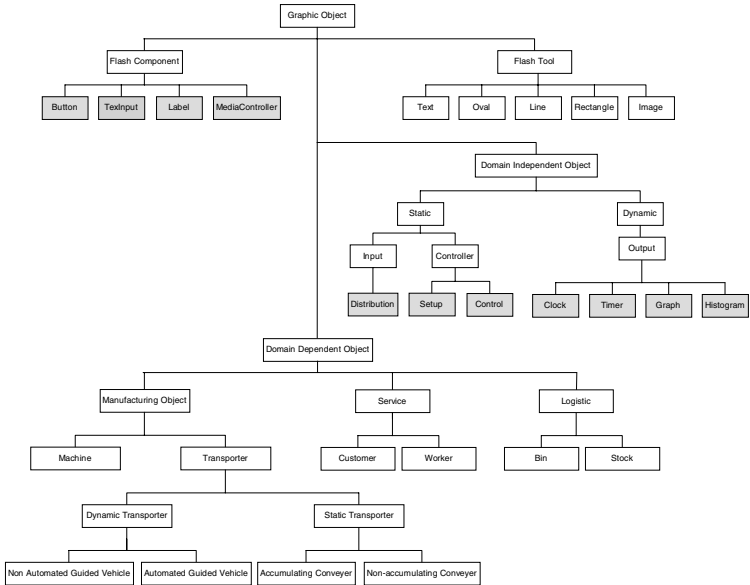


Fig. 5. Some *Graphical Objects* in Discrete Event Simulations

Domain independent graphical objects can be further divided into two subgroups: static objects and dynamic objects. *Static* objects do not move or change visual appearance during an animation; e.g. symbols for the simulation controller and sources and sinks for workload items. *Dynamic* graphical objects, on the other hand, change their appearance and/or location. This category includes clocks, timers, histograms and graphs.

Domain dependent graphical objects are often dynamic objects that represent *Sim-Processes* changing location (e.g. moving customers or vehicles) and/or appearance

(e.g. machines or conveyor belts). Figure 5 depicts some examples of domain dependent objects for manufacturing, service and logistic systems. To create such objects in Flash, their appearance must first be drawn or imported, converted to symbols and attached to a class that extends Flash's *MovieClip* class (e.g. a *SimProcess*).

4 Flash Components

Flash's components offer reusable functionality encapsulated in interfaces. To use a component a user needs to drag an instance onto a stage and change its default parameters, entering values (using a graphical interface) or writing some code (using its API). Flash lets developers create components by extending the *MovieClip* class. All Flash components are movie clips that are controlled by an ActionScript class ([19] and [20]); i.e. Actionscript classes must be coded to represent component behaviours. In order to customize properties and behaviours, components can be parameterized. This can happen either before or during an animation.

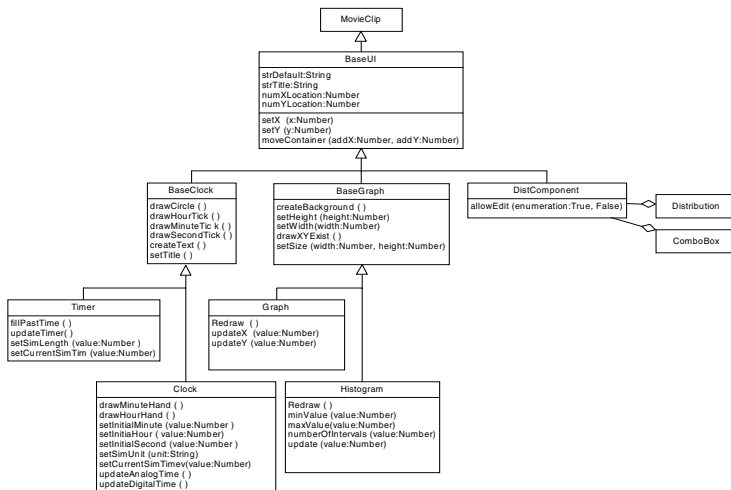


Fig. 6. Class Diagram of *Components* for Simulation Input and Output

A *MovieClip* is a generic animation object whose changes in visual appearance are defined on a timeline. *MovieClips* may contain graphics, audio or video, and can be nested recursively; i.e. clips inside clips, inside clips. A rapid succession of visual changes at run time creates animations. For example, a movie clip representing a customer in a bank simulation may move across a stage, from a source (door) to a server (teller), while a clip embedded inside it may play an animation (i.e. walking by moving arms and feet).

Movie clips are well suited for creating animation objects in DES. In addition to adding their own special features, component classes extend the *MovieClip* class to inherit all properties (such as location, visibility etc.) and methods (such as *createEmptyMovieClip()* and *attachMovie()*, etc.) of the *MovieClip* class. Figure 6 shows an example of the class diagram for components related to simulation input and output (under domain independent object in Figure 5).

5 Example

This section presents a simple example of how the DES libraries and components we have developed so far may be used to model a queueing scenario. Note that a basic familiarity with the Flash environment and some Actionscript coding is needed to use these tools. The example simulates a bank, where customers *arrive*, *walk* to a counter, get *served* by a teller and *exit* from the bank. The corresponding model uses a single *Server* object for the teller, a stream of *SimProcess* instances representing customers, and a number of distribution objects for sampling the duration of various model-time consuming activities.

To represent customers we must first create a new ActionScript class and save it under an appropriate name (in this case *Customer.as*) to the *simulation tools* folder. Here we define a *Customer* class based on *SimProcess*, declare various class variables and define a *lifecycle* method (see Listing 1). For example, in lines 7 to 10 we declare an *InterArrivalTime* distribution for sampling time delays between successive customer arrivals, a *WalkToCounterTime* distribution for sampling the time taken by customers to walk from an entry to a counter, a *ServiceTime* distribution to sample the time needed for a customer service, and a *WalkToExitTime* distribution for sampling walking time from counter to exit. In line 11 we declare a *teller* variable representing an object of the *server* class, and lines 12 to 15 declare variables for *Clock*, *Timer*, *Graph* and *Histogram* respectively. Note that names for these components (e.g. *myClock*, *myTimer* ...) must correspond to the names chosen during model initialization in the *Monitor* class (see Listing 2).

The *init* method (line 17) initializes *customer* objects. Here we must specify a sequence of phases (i.e. a *lifecycle*) all *Customers* instances step through. The *addPhase* method in line 18 attends to this requirement. The *lifecycle* method's description begins with updates to *Clock* and *Timer* components (lines 23 and 24) and continues with a description of what will happen when control returns to this object; based on the phase it is in (lines 27 to 54). Upon *arrival*, the first phase of the lifecycle (line 27), a customer object creates its successor with a call on the *createNew* method and advances itself to the next phase by calling *delay*. The call on *createNew* (line 28) instantiates a new *Customer* object, whose associated movie clip is then used to animate it on the stage. *delay* (line 30) schedules the current customer to continue to its next phase and inserts a corresponding event notice at the appropriate point on the agenda. At the right model time instant the monitor will later remove this event notice from the head of the agenda, retrieve the associated object and direct it to continue its execution from the relevant point on its lifecycle. The monitor will terminate the simulation when the end of the requested duration is reached or when no more events can be found on the agenda.

In preparation for the model's animated display, the *initLocation* method (line 31) initializes the stage location for arriving customer objects and the *MoveTo* method (e.g. in line 32) moves a customer's picture to a given location (e.g. that of a server entity). While the previously described actions prescribe simulation activities, these two methods serve animation. Note that *MoveTo* uses a motion tween, whose duration is controlled by the ratio of animation to simulation time, a value that can be dynamically adjusted by the viewer.

Server objects have two methods: *request* and *release*. *request* (line 35) allocates any free unit to a requesting customer. If all available capacity has been used, a customer object has to wait in a queue. A call on *release* (line 47) reactivates a customer object, returns however many capacity units it holds, and gives the next waiting customer a chance to acquire those units. In the final phase of a customer object's lifecycle, customers compute the time spent in the model and use this value to update a histogram (line 52). Invoking the *remove* method (line 53) destroys the customer object, whose storage will eventually be reclaimed by Flash's garbage collector.

Notice that we had to use a switch case statement to execute different sections of code, based on the *phase* a currently executing instance of the Customer class was in. *Phase*'s value was stored in a *phase* attribute and the *addPhase* method listed 6 valid phases. While this construction is arguably a rather clumsy way to implement a process oriented modeling framework, it was forced by ActionScript 2's lack of support for either coroutine, threads or any other control abstraction which would allow the persistence of state that could store one of multiple entry points to a method.

```

1  // import packages
2  import Monitors.*;
3  import Resources.Server;
4
5  class Customer extends SimProcess {
6      public static var Count:Number = 2;
7      public static var InterArrivalTime;
8      public static var WalkToCounterTime;
9      public static var ServiceTime;
10     public static var WalkToExitTime;
11     public static var teller;
12     public static var myClock;
13     public static var myTimer;
14     public static var myGraph;
15     public static var myHistogram;
16     public static var myEntry;
16     public static var myExit;
16     public static var myBench;
16
17     private function init ():Void {
18         addPhase("ARRIVAL,  ARRIVE_COUNTER,  SEIZE_TELLER,  DE-
LAY_TELLER,
19             RELEASE_TELLER,  DISPOSE");
20     }
21
22     public function lifeCycle (phase) {
23         myClock.currentSimulationTime = now();
24         myTimer.currentSimulationTime = now();
25
26         switch (phase) {
27             case "ARRIVAL":
28                 Customer.createNew("Customer", "Customer#" + Count++,
29                     Customer.InterArrivalTime.sample(), monitor);
30                 delay(Customer.WalkToCounterTime.sample());
31                 initLocation(myEntry);
32                 moveTo(myBench);
33                 break;
34             case "ARRIVE_COUNTER":
35                 teller.request(this);
36                 myGraph.updateX = now();

```

```

37         myGraph.updateY = teller.getQueueLength();
38         break;
39     case "SEIZE_TELLER":
40         delay(0);
41         moveTo(teller);
42         break;
43     case "DELAY_TELLER":
44         delay(Customer.ServiceTime.sample());
45         break;
46     case "RELEASE_TELLER":
47         teller.release();
48         delay(Customer.WalkToExitTime.sample());
49         moveTo(myExit);
50         break;
51     case "DISPOSE":
52         Customer.myHistogram.update = now() - getBirthTime();
53         remove(); // remove this object
54         break;
55     } //end switch
56 }
57 } // end Customer class

```

Listing 1. The Customer Class

In addition to customer objects, which arrive, request services and leave, we need to specify the environment these dynamic objects are to operate in; i.e. we need to add relevant components to Flash' stage and link these to the model's *Monitor* entity (see Listing 2). In lines 6 to 9 we initialize all global variables for the *Customer* class. Note that *myClock*, *myTimer*, *myGraph*, *myHistogram* and *myMonitor* must be the names we chose when placing the relevant components on the Flash stage; i.e. instances of classes *Clock*, *Timer*, *Graph*, *Histogram* and *Monitor* respectively. The *Monitor* component provides methods that specify simulation length (*simulateFor*), *animSpeed* for controlling animation speed, and *animate* to start animating a model. Animation speed or viewing ratio is the number of time units of model time per second of viewing time. For example, if the viewing ratio is 10, then the animation is running at a rate of 10 simulation time units per viewing second (real time). In line 17 we create a first customer object and schedule its execution at time 0.

```

1  // import package
2  import Monitors.SimProcess;
3
4
5
6  Customer.myClock = myClock;
7  Customer.myTimer = myTimer;
8  Customer.myGraph = myGraph;
9  Customer.myHistogram = myHistogram;
10
11 myTimerComp.simulationLength = _root.numSimLength;
12 teller.capacity = _root.numCapacity;
13 Customer.teller = teller;
14 myMonitor.simulateFor = _root.numSimLength;
15 myMonitor.animSpeed = _root.numAnimSpeed
16
17 Customer.createNew("Customer", "Customer#1", 0, myMonitor);
18 myMonitor.animate();

```

Listing 2. Simulation *Model* Code

To complete our model's definition and use the above *Customer* and *Model* classes, we must first create a new Flash document. For this example we need just two keyframes: *Parameter* and *Animation*.

The *Parameter* keyframe displays a form for choosing statistical distributions and their parameters, server capacities and length of a simulation run. Relevant components are dragged from the Components Panel and dropped at appropriate places on the Flash stage. Note that each component has to be given a unique identifier that corresponds to the names used by the *Monitor* class (see Listing 2).

The *Animation* keyframe is used as a stage to assemble the visual representation of the model's animation. Here we use simulation components, such as Clock, Timer, Graph and Histogram, whose properties (e.g. colour, width, height, etc.) can be changed through a Component Inspector. For each change in properties the component's appearance on the stage will be automatically adjusted. The model layout itself can either be drawn using Flash's drawing tools, or we can import external graphic files in JPEG or DXF formats.

To animate customer and server objects we must attach visual images. Such images must first be converted to *MovieClips* and then attached to *Customer* and *Server* classes. Once the movie clip is in the Flash library we can define the customer objects' visual appearance based on keyframes named *onMoving*, *inQueue* and *inProcess*. The server object can be animated in a similar way, i.e. by assigning different representations to keyframes *Idle* and *Busy*. Note that these frames are defined on the *Customer* symbol's timeline and *not* globally on the stage. This gives us a local animation for customers (i.e. their change of appearance in different states) that is nested inside the main animation (tweened movements across the stage). Figure 7 shows a snapshot of an animation of the example.

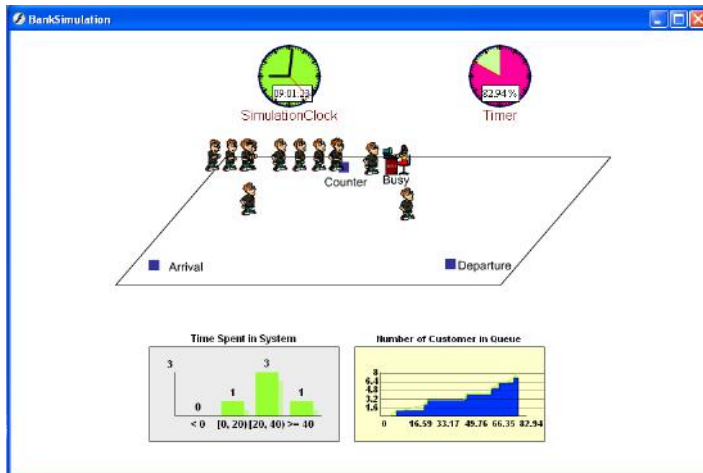


Fig. 7. Animation Output

6 Conclusions

Adobe Flash offers a good platform for developing interactive, web-based and multi-media-enriched simulation libraries and components. By sub-classing its *MovieClip* class developers of eLearning materials can easily customize animated discrete event simulation scenarios and control the visual appearance through a mixture of drag-and-drop-style symbol manipulation and ActionScript coding. Since Flash also provides good support for multiple media, such as text, sound, video, and animated graphics, simulations can be made to come “alive” and attract learners’ attention and interest. During run time the Flash player allows users to stop, make changes and resume a simulation, as well as zoom in, zoom out and pan around the stage. In addition, our toolbox permits simulation and animation speeds to be dynamically adjusted, so that learners can focus at leisure on interesting locations and trace interesting sequences of events.

We are continuing research into building highly interactive and visual environments for constructing and animating Flash-based simulation models. The current need to annotate all lifecycles of dynamic objects with phase information (see Listing 1) is not as well suited to occasional users, such as teachers developing eLearning materials, as we would wish it to be. This is due to the lack of suitable semantic abstractions for providing a *coroutine* feature. Ideally there should be no need for Actionscript coding at all, so that models and animations could both be constructed by dropping and linking components from libraries while cloaking them in appropriate visual representations. Unfortunately Actionscript currently offers no support for turning text into code (i.e. there is no equivalent to an *eval* statement) and a small compiler would need to be written to allow users the flexibility to alter dynamic components’ behaviour through visual interfaces. We have started to look at alternative architectures to circumvent this restriction. For example, simulation events could possibly be attached to key frames on Flash’ timeline. In this fashion an animation describing an entity’s visual transformations along its timeline would be in charge of describing the dynamics of both model (i.e. changes in the entity’s abstract state) and animation (i.e. changes in the entity’s appearance and location).

Components and libraries for other domains, such as manufacturing and logistic systems are also under development, as are empirical investigations of what tools may best help teachers ease the assembly of component-based simulations and embed the resulting models in a learning management system.

Acknowledgements. The original cartoon movie clip used to represent animated customers in the simulation was taken from <http://www.kirupa.com> and edited to represent their transient behaviour in the model.

References

1. Banks, J.: Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice. John Wiley & Sons, Chichester (1998)
2. Kelton, W.D., Sadowski, R.P., Sturrock, D.T.: Simulation with Arena, 3rd edn. Mc-Graw Hill, New York (2004)

3. Syrjakow, M., Berdux, J., Szdzerbicka, H.: Interactive Web-based Animations for Teaching and Learning. In: Proceedings of the 2000 Winter Simulation Conference, Orlando, Florida, pp. 1651–1659 (2000)
4. Rosson, M.B., Seals, C.: Teachers as Simulation Programmers: Minimalist Learning and Reuse. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Seattle, Washington, pp. 237–244 (2001)
5. Smialek, M.: Developing e-Learning Simulations With Tools You Already Know. The e-Learning Developers Journal (2002) (Retrieved May 12, 2008), <http://www.elearningguild.com/pdf/2/120302DEV-P.pdf>
6. Mohler, J.L.: Flash 8: Graphics, Animation and Interactivity. Thompson Delmar Learning, New York (2006)
7. Shupe, R., Hoekman, J.R.: Flash 8: Projects for Learning Animation and Interactivity. O'Reilly Media Inc., Sebastopol (2006)
8. Peters, K.: Foundation ActionScript 3.0 Animation: Making Things Move. Friends of ED, Berkeley (2007)
9. Peters, K., Yard, T.: Extending Macromedia Flash MX 2004: Complete Guide and Reference to JavaScript Flash. Friend of ED, Birmingham (2004)
10. Sanders, W.B.: Macromedia Flash MX Professional 2004: Kick Start. Sams, Indianapolis (2004)
11. Lopez, L.A.: New Perspectives on Macromedia Flash 8: Comprehensive. Thompson Course Technology, Boston (2006)
12. Kuljis, J., Paul, R.J.: A Review of Web Based Simulation: Whither We Wander? In: Proceedings of the 2000 Winter Simulation Conference, Orlando, Florida, pp. 1872–1881 (2000)
13. Garrido, J.M.: Object-oriented Discrete-event Simulation: A Practical Introduction. Kluwer Academic/Plenum Publishers, New York (2001)
14. Page, B., Kreutzer, W.: Simulating Discrete Event Systems with UML and Java. Shaker Verlag, Aachen (2005)
15. Kaye, J.M., Castillo, D.: Flash MX for Interactive Simulation. Thompson Delmar Learning, New York (2003)
16. Moock, C.: Essential ActionScript 2. Farnham, O'Reilly (2004)
17. Birtwistle, G.M.: DEMOS: A Discrete Event Modelling on Simulation. McMillan, London (1979)
18. Rossetti, M.D., Aylor, B., Jacoby, R., Prorock, A., White, A.: SIMFONE: An Object-Oriented Simulation Framework. In: Proceedings of the 2000 Winter Simulation Conference, Orlando, Florida, pp. 1855–1864 (2000)
19. Hamlin, J.S., Tarbell, J.: Williams: The Hidden Power of Flash Components. Sybex, San Francisco (2003)
20. Antonio, D.D.: Advanced ActionScript Components: Mastering the Flash Component Architecture. Apress, Berkeley (2006)

Preservation of Client Credentials Within Online Multi-user Virtual Environments Used as Learning Spaces

Sharon Griffith and Ioakim Marmaridis

University of Western Sydney, Sydney, Australia

Abstract. Online Multi-User Virtual Environments such as Second Life have become a popular educational resource when creating learning spaces for the net-enabled generation of students. High expectations within these environments for breadth of functionality and high speed of content delivery, place a heavy demand on systems integration between the MUVE and web based learning tools. The Sloodle project attempts to combines the Second Life MUVE and the Moodle LMS. This paper highlights the considerations for successful client accreditation from Sloodle to another MUVE called OpenSimulator and more generally shows an architecture for cross-authentication between numerous of MUVEs. This blend of proprietary and open source systems and their integration will highlight the current limitations and future development of cross authentication of clients (students and educators) between traditional LMS, virtual classroom tools and MUVEs.

1 Introduction

The nature of our interconnected world poses challenges for education and puts demands for new methods of delivering learning materials. MUVEs as a teaching platform are gaining in popularity as people see their value in education as a media rich and interactive platform for teaching and learning. Using Second Life in collaboration with Moodle has enabled students to experience a blended learning environment [1]. Students are encouraged to work collaboratively, forming learning communities where each participant is both a teacher as well as a learner [2]. Many educators who are not familiar with MUVEs are also exploring these environments with no prior experience in MUVE technology [3].

There is a definite move to integrate traditional tools for e-learning with MUVEs, and one successful application is the virtual space Sloodle. The LMS Moodle enables users to participate in remote web-based learning and supplements traditional classroom learning. It provides teachers and trainers a powerful set of web-based tools for a flexible array of activities, including assignments, forums, journals, quizzes, surveys, chat rooms, and workshops [4]. As an educational facility, Moodle is guided by a social constructionist pedagogy with an emphasis on tools that promote collaboration and self-evaluation [5].

One of the key concerns when moving between virtual spaces is the preservation of credentials within online Multi-User Virtual Environments. This can

be achieved via a mechanism for cross-authentication of the user from in-world avatar to user of the LMS. This needs to be performed correctly because of the potentially sensitive data recorded about the user in both environments.

This paper describes the current approaches in use for cross-authenticating users between virtual spaces. User activities between the MUVE and traditional web-based tools are reviewed, with emphasis on the learning environment of Sloodle as a concrete example of how these authentication techniques have been implemented. It builds on this work to suggest an authentication framework to maintain user credentials into other MUVEs such as OpenSimulator.

2 Background and Motivation

In this field of research a number of terms exist that are both specific and relevant and are required to be explained up front. The most commonly used terms are below along with some explanation of their origin.

The term online multi-user virtual environment (MUVE) was first discussed by Morningstar [6] in the paper titled "The Lessons of Lucasfilms Habitat". The attempt to create a very large scale commercial MUVE using a low end computer with avatars connected to the central system via a packet switched network enabled the formation of graphical online virtual environments to become a commercial reality. The term has been extended to now include the popular phrases of online spaces including that of persistent virtual metaverses. A MUVE does not revolve around gameplay as the primary activity, like most Multi-User Dungeons (MUDs), Object-oriented MUDs (MOOs), Massively Multiplayer Online Role-Playing Game (MMORPGs) or other Massively Multiplayer Online Game (MMOGs). Instead its focus is to simulate real world interactions between people; in environments similar to the real world, a fantasy world or a hybrid combination.

The metaverse concept was first introduced by Neal Stephenson's 1992 science fiction novel "Snow Crash". It details how humans as avatars interact with other avatars (human and software agents) in a 3D virtual space representing the real world.

Second Life or more commonly known as SL, is a persistent 3D MUVE developed by Linden Research Inc. Since its 2003 release, Second Life has become a popular tool for social networking, online education as well as offering in-world services for trade, business management and entertainment [7]. Users in Second Life are known as Residents who are able to construct 3D objects (prims) and create scripts via the Linden Scripting Language (LSL) allowing for interactivity in-world. Connectivity between in-world and real world applications are created via prims using the LSL. It is this connectivity between a MUVE and learning management system which we will explore further.

Learning management systems (LMS) are a category of software system capable of delivering training and learning materials via an online medium such as the World Wide Web (web). The degree of complexity within the LMS can vary according to the package type chosen and its intended use. Most LMS are

web-based, and can be accessed via a web browser. An LMS typically houses content that is either commercially obtained or user created. One of the popular features of a LMS is its ability to offer templates and simple forms to create interactive web-based class environments [8]. Due to the nature of online learning, both commercial and open source LMS offer learning institutions the ability to choose a software package based on their requirements and budget.

The open source LMS Moodle is a software package designed using sound ideological principles; to help educators create effective online learning communities. This e-learning software platform features a modular design which enables people to develop additional functionality. Its development is conducted by both commercial and non-commercial users worldwide [9].

As an open source project, Sloodle integrates the MUVE of Second Life with the Moodle learning-management system [10]. It provides a variety of tools to support learning and teaching in the immersive virtual world. These tools are fully integrated with the successful web-based Moodle LMS which enables many educational institutions world-wide to engage in a new online collaborative space.

Our motivation for this work is to provide an overview of the existing approaches for achieving cross-authentication between Second Life and Sloodle. We also propose a high-level architecture for cross-authentication across a number of Metaverses currently available, including Linden Research Inc - Second Life, Google's - Lively, Makena Technologies - There, the community-driven OpenSimulator and more.

3 Existing Approaches and Considerations

One of the most well known projects in the area of integration between Metaverses and traditional e-learning tools is Sloodle. It combines the popular Moodle learning management system (LMS) with the Second Life Metaverse. Figure 10 shows how conceptually the Moodle LMS and Second Life Metaverse come together to compliment each other and provide a new dimension of learning through a configurable number of integration points offering two-way interactions from Moodle to the Metaverse and vice versa.

To achieve this level of integration, Sloodle had to provide a mechanism for authenticating users between the two different systems and maintain their identity as they moved between them during the normal course of learning interactions. The commonly used terminology in Sloodle for maintaining user credentials across the two systems is "Registration". "Registration" is otherwise known as "authentication" or "linking", and it is this process which links a Second Life avatar to a Moodle account. The Moodle site does this by storing a list of avatars it has interacted with, and keeping a note of the Moodle account owned by the same person. This is done so that, when a user in Second Life interacts with the "in-world" environment, their details can be stored on the Moodle site as well directly as common Moodle user [10].

There are two modes of registration, manual and automatic. Each mode is briefly explained below.

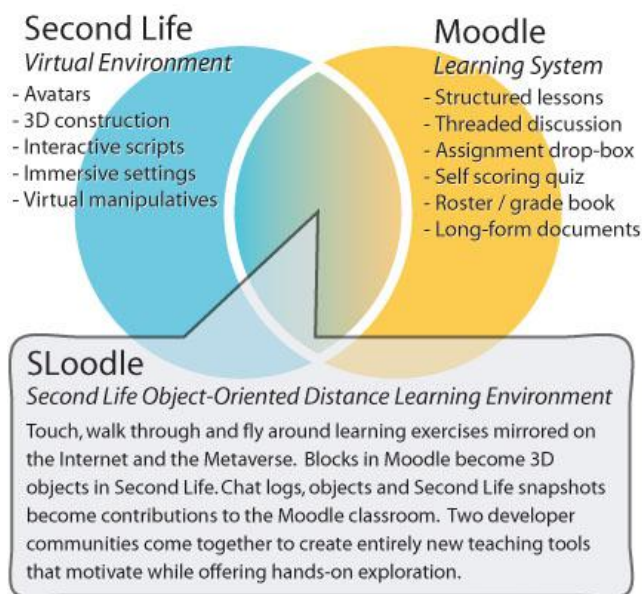


Fig. 1. Sloodle Conceptual Diagram - One of the approaches to integrating Second Life and Moodle

Manual registration - Objects can perform 'manual registration', where you are given a URL in Second Life, and must use it to login to the Moodle site. This is best used where users are likely to have a Moodle account before they have an avatar, and where Moodle is likely to be used substantially alongside Second Life.

Auto registration - When an avatar in Second Life attempts to use one of your Sloodle tools, Sloodle will in turn check to see if the avatar is already registered in the Moodle LMS. If not, it will attempt to automatically create a new Moodle account for them, under the avatar's name. This process is completed almost instantly, allowing the avatar to immediately begin using the tool, as if they were already registered. Auto registration greatly simplifies the initial process of linking an avatar to a Moodle user account, and it reduces the number of objects you need to have configured in Second Life. It is useful if you do not expect or require your users to use Moodle directly in their web-browser. It is therefore well suited in situations where users will have avatars before having Moodle accounts.

Other techniques for user authentication within the Sloodle environment are:

- Prim passwords - These are internal passwords placed upon objects in Second Life for use in Moodle. The password is always numeric, between 5-9 digits long. Whenever a Sloodle object wants to communicate with Moodle at all, it must supply this password.

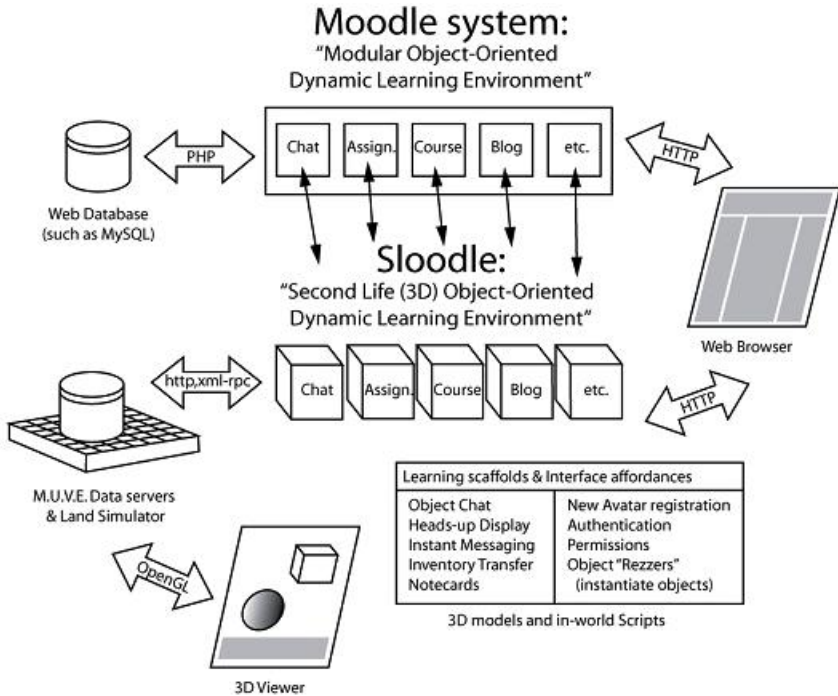


Fig. 2. Sloodle overview diagram - integration between Moodle and Second Life

- Object authorisation - Object authorisation is an alternative to the Prim password security method for Sloodle objects. Instead of having a single password for all objects in the course/site to use, the teacher or administrator must explicitly authorise an individual object whenever it is going to be used. If the object's key (UUID) changes (e.g. duplication, deletion etc.), then it must be re-authorised. As each object in Second Life has a UUID, it can be used for the authentication process by recording each UUID against an asset database stored on Second Life asset servers. The cross authentication process is then applied between objects in-world to real world servers offering web-based services.

Figure 2 from Livingstone and Kemp [3] shows how Moodle and Second Life are brought together via Sloodle.

4 Limitations of Current Approaches

The Sloodle implementation exhibits many limitations, these are discussed below:

- Under auto registration, the system will attempt to send the avatar an instant message in Second Life, containing their new username and password

for Moodle; but this notification may not arrive for various reasons. To counteract this problem, one can rez (create in SL-speak) a password reset tool – when an auto-registered avatar touches it, their Moodle password will be reset to something random, and immediately reported privately back to them in Second Life. For security reasons, this device can only be used if the user has never logged-in to Moodle before. Auto registration will always create new Moodle accounts on the local system (so it is not compatible with external authentication databases). If you want to change the Moodle account an avatar is linked to, then you may use the Sloodle User Management tools.

- Sloodle offers a configuration page, which can only be accessed by Moodle administrators, must be used to enable/disable auto registration for the whole site. If auto registration is disabled here, then it cannot be used anywhere on the site. This is the default initial setting, because not all Moodle sites will be compatible with this option; if using an external authentication database for instance. After auto registration is enabled on the site, it must also be enabled on each individual course where the educator would like to use it. This can be done through the Sloodle Course Settings page, accessible via the optional Sloodle Menu Block, or through a link on a Sloodle Controller Module page.
- Prim passwords are only numeric with a character length of 5-9 digits. This password must be supplied when a Sloodle object needs to communicate with a Moodle object. This process is inefficient, as the users of the system are required to manually enter this information during this transaction.
- Object authorisation requires a repository to keep track of multiple passwords. Also, UUID changes to objects require a password change. This behavior, combined by lack of automation, makes for a rather inefficient authorisation method.

5 Proposed Approach

5.1 Interoperability of MUVES

The ability to successfully communicate between MUVE's lead to both IBM and Linden Research Inc announcing that their research teams had successfully teleported avatars from the Second Life Preview Grid into another MUVE running on an OpenSim server in July 2008 [11]. This achievement marked the first time an avatar has moved from one MUVE to another. Its an important first step toward enabling avatars to pass freely between MUVES, since the formation of the Architecture Working Group (AWG) [12] in September 2007. The Open Grid Protocol used in the project enables interoperability between virtual worlds. This first step demonstrated the interconnectivity between Second Life with other MUVES, but also the ability to link other MUVES with one another. An open standard for interoperability based on the Open Grid Protocol would allow users to cross freely from one MUVE to another, similar to how people traverse from one website to another via the web.

The result of this project only allowed the avatars to login and teleport between the MUVES. Inventory assets, such as skins or objects were non transferable [11]. In addition to the experiment by IBM and Linden Research Inc, the complete transition from one MUVE to another whilst retaining the clients assets, requires further research and development between both proprietary MUVE developers and the open source community. Using the foundation experiment from that project we highlight future areas for consideration when engaging in the cross-authentication process.

Figure 3 shows how the proposed architecture for the cross-authentication system looks like:

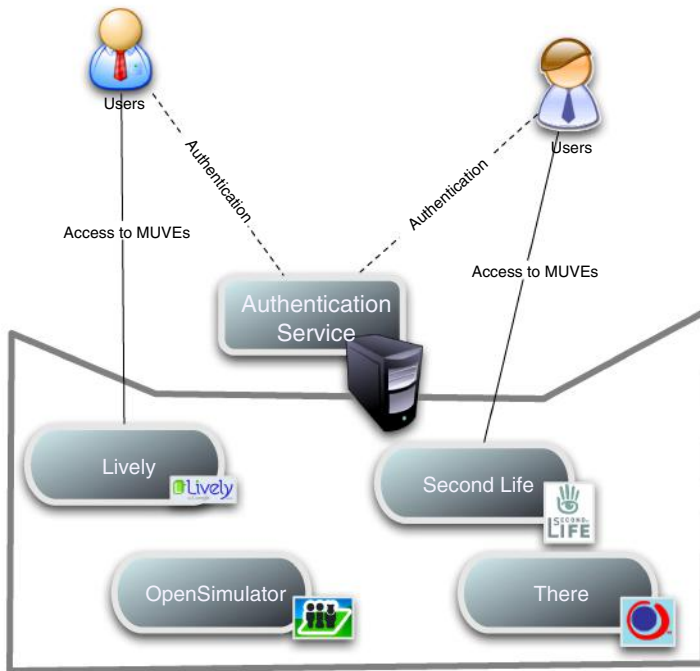


Fig. 3. Cross-authentication between MUVES - high-level architecture diagram

The core idea is to logically centralise authentication services so that only one interface has to be created for each MUVE. This is unlike the current experiment between OpenSimulator and Second Life where changes to both environments had to be made and co-ordinated in order for the cross-authentication to work. By logically centralising the authentication developers of other MUVES should be able to offer similar abilities for their avatars to enter other MUVES as well as accept remove avatars into their worlds.

A design of such system for cross-authentication must also meet a number of criteria, namely:

- Designed for security where actions can be tracked and traced and repeated unsuccessful attempts for authentication trigger defense mechanisms.
- Designed for reliability and to withstand failure of individual components both at the physical hardware level and within the software design itself.
- Easy setting of permissions while working in a group environment and collaborating with other students and/or educators.
- Make creation and maintenance of accounts in the individual systems as automated and transparent as possible.
- Be extensible so as to integrate with future systems.
- Open, well documented protocol that is easily implementable.
- Use a well-understood transport; particularly one compatible with web technologies (XMLRPC or HTTP GET/POST based, RESTful APIs etc).
- Decoupled, pluggable design that allows for authentication middleware to be re-implemented without changing the API details and while keeping compatibility of interfaces to the individual MUEs.

6 Benefits of the Approach

It is evident that as virtual worlds gain in popularity and offer increasingly more services, people will want freedom of movement between worlds as they require freedom of expression within those worlds. To that end, there are substantial benefits flowing from establishing and following a high-level architecture to the approach used for cross-authentication of avatars between MUEs.

These benefits in the realm of education include the following:

- Interoperability between virtual world spaces and current e-learning technologies for new pedagogical educational practices. These spaces can supplement traditional learning spaces, with the integration of new platforms which deliver interactive content via the Internet.
- Consistent framework for authentication will allow additional learning tools and applications to be added to the ecosystem and become rapidly available.
- Contributing to the open source community via promotion of open standards and knowledge transfer, attracting new people and fresh ideas in education.
- Attract students to using the integrated ecosystem by lowering the barrier of entry and allowing easier transparency between the different facilities available.
- Introduce transparency at the technology layer allowing educators and students higher freedom of expression with the tools available.

The listing of these items are by no way exhaustive. As the open standard for MUE interoperability and open protocol grid matures, existing and new technologies will proliferate enabling users to choose a teaching and learning platform suitable to their needs and better still, to mix and match best of breed applications between numerous MUEs and LMS environments.

7 Conclusions and Future Work

Multi-user virtual environments are now becoming the next platform for delivering interactive content to users. Although, originally designed for an audience compatible with the online games market, educators are now using these platforms to attract the net-enabled generation of users for teaching and learning high school/University curriculum. The cross pollination of these technologies requires a stable and robust authentication process. As the interactivity between multiple MUEs and the LMS engage more users. We see the work presented in this paper as a step towards this direction, raising awareness of the current state of play and putting up-front the requirements for open standards and extensible architectures that can facilitate growth of inter MUE and LMS use.

Many interesting areas of research are opened up by the opportunities that lay ahead of us. These include the study of how to successfully expand the recommended authentication architecture and framework to incorporate more web-enabled systems that can offer services for educators and students. Also, there is work that must be done into supporting the proposed authentication architecture and mechanism over mobile devices for "just-in-time" (JIT) learning. Investigating collaborative and educational tools is a very exciting field to be working in and we hope that many others will continue to see the potential that exists in interconnecting tools and systems in an open, transparent way with high security and reliability.

References

1. Hetherington, R., Bonar-Law, J., Fleet, T., Parkinson, L.: Learning in a Multi-User Virtual Environment. In: Visualisation, 2008 International Conference, pp. 99–105 (2008)
2. Kluge, S., Riley, L.: Teaching in Virtual Worlds: Opportunities and Challenges. Issues in Informing Science and Information Technology 5 (2008)
3. Livingstone, D., Kemp, J.: Integrating web-based and 3d learning environments: Second life meets moodle. Next Generation Technology-Enhanced Learning, 8
4. Jason, C., Helen, F.: Using moodle, 2nd edn. O'Reilly, Sebastopol (2007)
5. Corich, S.: Is it time to Moodle? In: Proceedings of the Eighteenth Annual Conference of the National Advisory Committee on Computing Qualifications. NACCQ, Tauranga (2005)
6. Morningstar, C., Farmer, F.: The Lessons of Lucasfilm's Habitat. Cyberspace: First Steps, 273–301 (1991)
7. Wikipedia, Second life - wikipedia (November 2008), http://en.wikipedia.org/wiki/Second_Life
8. Kemp, J., Livingstone, D.: Putting a Second Life Metaverse skin on learning management systems. In: Second Life Education Workshop at the Second Life Community Convention, San Francisco, August 20 (2006)
9. Wikipedia, Moodle - wikipedia (November 2008), <http://en.wikipedia.org/wiki/Moodle>

10. Sloodle, Sloodle - virtual environment learning system (November 2008),
<http://www.sloodle.org/moodle/>
11. Linden, H.: IBM and linden lab interoperability announcement (July 2008),
<http://blog.secondlife.com/2008/07/08/ibm-linden-lab-interoperability-announcement/>
12. LindenResearchInc., Architecture working group - second life wiki (November 2008),
http://wiki.secondlife.com/wiki/Architecture_Working_Group

Short Term Stock Prediction Using SOM

Prompong Sugunsil¹ and Samerkae Somhom²

¹ Department of Computer Science,
Faculty of Science, ChiangMai University,
ChiangMai 50200, Thailand
g500531139@cm.edu,

² Department of Computer Science,
Faculty of Science, ChiangMai University,
ChiangMai 50200, Thailand
samerkae@chiangmai.ac.th

Abstract. In this paper, we propose a stock movement prediction model using self organization map. The correlation is adapted to select inputs from technical indexes. The self-organization map is utilized to make decision of stock selling or buying. The proposed model is tested on the Microsoft and General Electric. Through the experimental test, the method has correctly predicted the movement of stock with close to 90% accuracy in trainnig dataset and 75% accuracy in datatest. The results can be further improved for higher accuracy.

Keywords: Neural network, self-organization map, stock market, stock prediction.

1 Introduction

Nowadays, the capital of the country is heavily linked with the stock market. The prosperity of the stock market also means the wealth of nation. The stock market has high liquidity that investor can quickly and easily change the asset to cash. Although the investment in stock market is attractive, there is a small number of people investing in the stock market because it is uneasy to predict behaviors. The intrinsic property of the stock market is complexity. It is characterized by nonlinearity, uncertainty, data intensity, noise and non-stationary [1]. Even a simple factor could cause a tremendous effect to the stock market.

The variation of stock price is an interesting subject. Two basic analytical approaches are fundamental analysis and technical analysis. Fundamental analysis takes economic factors such as industrial production, inflation, risk premium, consumption, oil prices, statistical macro variable, etc. [2] in account. Technical analysis is based on financial time series data generated from daily trading data. Nevertheless, it is difficult to determine the most appropriate time to buy, hold or sell stock.

Short term stock prediction is difficult because there are various factors that influence stock prices and each factor has different effects at a time. However, the factors influence over a period of time. In the same period, if we could model the behavior of stock, we should predict the movement of stock.

Recently, many approaches has been applied to predict stock price and movement, They include neural network, Bayesian belief network, genetic algorithm, fuzzy set and hybrid method.

In [3], self-organizing map were proposed and showed notable success. The proposed system consumed the continuous trading data as input. In this paper, we propose the extension of the system presented in by changing the input from trading data to technical indexes.

2 Self-Organizing Map

A self-organizing map is an unsupervised neural network model. It consists of two layers which are input layer and output layer. The input layer has node as much as the number of pre-assigned input variable. The output layer or weight vector represents the exemplar of the input.

Through the self organizing process, the output neuron of output layer of which weight vector matched the input pattern most closely is chosen to be the winner. The winning unit and its neighboring unit update their weights[4]. This process continues until stopping condition becomes true.

In matching step, the winning node $i(x)$ is determined by using the minimum distance Euclidean criterion:

$$i(x) = \operatorname{argmin}_i \|x(t) - w_j(t)\| \quad (1)$$

where w_j denotes the synaptic weight vector of neuron j at time t . For the updating process, the synaptic weight vectors of all neurons were adjusted using the update formula:

$$w_j(t+1) = w_j(t) + \eta h_{j,i(x)}(x_i(t) - w_j(t)) \quad (2)$$

$$h_{j,i(x)} = \left(\frac{-d_{j,i(x)}^2}{2\sigma^2} \right) \quad (3)$$

where $x_i(t)$ is the input to node i at time t , $w_j(t)$ is the weight from input node i to output node j at time t . η is the learning rate parameter. $h_{j,i(x)}$ is the neighborhood function centered around the winning node $i(x)$. $d_{j,i(x)}^2$ is the distance between the winning neuron $i(x)$ and the adjacent neuron j . σ is the width of the topological neighborhood[5].

3 The Proposed Method

Our aim is to determine whether the time is appropriate to buy, hold or sell based on our prediction modeled from previous periods. We use short-period data for each modeling in order to model the behavior of stock under the same circumstance. When the model guides that the stock price will increase, we should buy. And when the model guides that the way of the stock price will decrease, we should sell or hold. We use self-organizing map as prediction method.

3.1 Preprocessing

We have a database full of continuous daily trading data. Each record consists of the opening price, the highest price, the lowest price, the closing price and the trading volume.

As in [6], we generate percent of change for each record from the opening price using the formula:

$$c(t+1) = \frac{f(t+1) - f(t)}{f(t)} \times 100 \quad (4)$$

where $c(t+1)$ is percent of change at day $t+1$, $f(t)$ is the data, the opening price, at day t .

We do not directly use the trading data as the input variable but we rather transform them into technical indexes being used by financial expert.

After the technical indexes is generated, each of them is normalized by calculating the ratio of value and range of the technical index in the same time frame. The normalization of data will increase efficiency of the prediction because the technical index has a lot of differences in range. We describe the formula as follow:

$$x_{new}^i(t) = \frac{x_{ori}^i(t) - \min(i)}{\max(i) - \min(i)} \quad (5)$$

where $x_{new}^i(t)$ is original data, $x_{new}^i(t)$ is normalized data at record t of technical index i , $\max(i)$ and $\min(i)$ s the maximum value and minimum value of technical index i in the same time frame.

3.2 Clustering Analysis

After data preprocessing, we employ K -means algorithm as clustering method on percent of change. K -means is a well-known clustering algorithm by which the data is partitioned the data into three groups that the sum of distance from data to the assigned group's mean point is minimum. We cluster the percentage of change into three groups which represent the periods for holding, selling or buying.

After K -means algorithm is complete, we group the hold period and the sell period together in order to distinguish between the interesting group and uninteresting group.

3.3 Self-Organizing Map

This step is to construct weight being used to determine whether the testing data is interesting or un-interesting by using self-organizing map. From previous step, we have groups assigned to each percentage of change.

We use a self-organizing map consisting of $n \times 2$ nodes to construct weight for n input variables and two groups, interesting and un-interesting. In weight updating process, we have to modify the update formula (Equa. 2) because we already have exemplar of the data, the interesting group and the un-interesting group. Our purpose is to construct weight used in prediction of future stock movement.

For weight updating process, the synaptic weight vectors were adjusted by using the update formula :

$$W_{kj} = \frac{\sum_{group(\alpha)=k \& \beta=j} x_{\alpha\beta} \times w_{k\beta}}{N_k} \quad (6)$$

$$w_{kj}(t+1) = w_{kj}(t) + \eta W_{kj} \quad (7)$$

where k is the label of weight vector representing the groups and j is the label of input node, $w_{kj}(t)$ is the weight from input node j to output node k at time t . η is the learning rate parameter. N_k is the number of record assigned with weight vector k . W_{kj} is the average direction of weight from output node k and input node j .

We stop the learning process when the self-organizing map becomes stabilized indicating by change of weight. When weight becomes stable, the change of weight also becomes smaller.

4 Experimental Results

We received Microsoft and General Electronic from YAHOO¹ as dataset. The data were reported daily from 14 September 2005 to 13 February 2008 total 609 records. We used data in the same time frame as the training dataset and the next record as the testing data.

We used learning rate parameter of 0.01 and initial weights of 0.01. We used 39 technical indexes such as acceleration, accumulation/distribution, advance/decline line, momentum, RSI, etc. We conducted the experiment to find the most appropriate value of parameter, size of time frame, that caused the method to produce the most accurate prediction result. The results are listed as follow:

From the accuracy of training dataset's prediction result of both Microsoft and General Electronic, the results show that as the size of time frame decreases, the accuracy increases. The best result is achieved when the number of day is 7. The result of training dataset is closed to 90% accuracy in 7 days. The best result of testing dataset is closed to 75% accuracy.

¹ <http://quote.yahoo.com>

Table 1. Average accuracy of Microsoft dataset prediction

Time frame (day)	Training dataset(%)	Testing dataset(%)
7	91.47	75.15
15	80.71	69.60
30	70.69	63.03
45	63.29	45.45
60	61.53	31.25
75	61.15	54.54

Table 2. Average accuracy of General Electronic dataset prediction

Time frame (day)	Training dataset(%)	Testing dataset(%)
7	92.64	74.45
15	85.00	66.77
30	83.79	65.86
45	87.41	54.55
60	73.05	53.84
75	81.59	45.45

5 Conclusion

In this paper, we proposed a self-organizing map for stock movement prediction. The system was tested on Microsoft and General Electronic. Although the performances of the proposed method in are notable in training dataset, the accuracy of prediction of testing dataset is significantly worse than testing dataset. The reason of testing dataset's may be the property of stock market itself, complexity. In addition, this approach is not only restricted to the stock market but it can be applied to other times-series data.

This is just the beginning of research. The long-term goal is to find a method that can predict the stock trend based on technical indexes with high accuracy.

References

1. Hall, J.W.: Adaptive selection of US stocks with neural nets. In: Deboeck, G.J. (ed.) Trading on the edge: neural, genetic and fuzzy systems for chaotic financial markets, pp. 45–65. Wiley, New York (1994)
2. Chen, N., Roll, R., Ross, S.A.: Economic Force and the Stock Market. *Journal of Business* 59(3), 383–403 (1986)
3. Zorrin, A.: Stock market prediction: Kohonen versus Back propagation. In: Proceedings of International Conference on Modeling and Simulation of Business System, Vilnius, Lithuania, pp. 115–119 (2003)

4. Fausett, L.: Fundamentals of Neural Networks. Architectures, algorithms and applications, pp. 169–187. Prentice Hall, Inc., Englewood Cliffs (1994)
5. Haykin, S.: Neural Networks: A comprehensive Foundation. Prentice Hall, New Jersey (1999)
6. Kwon, Y., Moon, B.: Daily Stock Prediction Using Neuro-Genetic Hybrids. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) GECCO 2003. LNCS, vol. 2724, pp. 2203–2214. Springer, Heidelberg (2003)

Rules and Patterns for Website Orchestration

René Noack

Christian Albrechts University Kiel, Department of Computer Science,
Kiel, Germany

`noack@is.informatik.uni-kiel.de`

Summary. Information-intensive websites are constantly changing. It concerns the enhancement as well as the maintenance of existing content, also layout changes are common practice. Changes often imply the need to reorganise some content parts to keep an adequate representation. At present, such changes will be performed statically, changing the rules of presentation regarding each content object. However, it is possible to generalise the content presentation in the way that layout definitions and restrictions determine the representation of content objects or their types. Therefore, this paper proposes dynamic placement and representation of content objects to avoid manual customisations and grave errors in layout development. The approach is based on patterns that allow to change and adapt the layout as well as help to reuse components and concepts.

1 Introduction

Information-intensive websites are constantly changing. Unfortunately, changes often result in presentation problems and effort is necessary to solve these. At present, the most of these changes have to be detected and solved manually. We notice that it is too time-consuming in the context of information-intensive websites. Therefore, we are searching for dynamic adaptation methods and concepts to reduce the maintenance effort.

Changes can have several reasons, wherefore we intend to take into account six dimensions of WIS development (intention, story, context, content, functionality and presentation) as introduced in [12]. These dimensions try to generalise development aspects with the aim of making the layout development process more flexible. Because of relations between the dimensions, a weighting factor was assigned to each dimension that depends on the application type. The weighting influences the execution order of the dimensions so that a development can result in different representations. Thus, it increases the flexibility of the whole layout development process as introduced in [10].

Changes against all development dimensions can result in layout changes. Layout changes concern the placement that can be defined as an assignment of content to a specified action space. As a consequence of assignments, placement problems are possible so that revisions become essential. Content over- and underflows are characteristic problems that can be dissolved either by layout or

by content restructuring. It depends on constraints as well as the application domain, which method can be seen as most appropriated.

Placement problems often occur if a layout has not a simple structure like very common two- or three-column grids. However to avoid problems, grids, as a basis for placement, should not be minimized at the expense of attractiveness because there is no guarantee that simple grids don't have any problem. Moreover, such a minimisation tries to circumvent existing problems and downgrades the usability as well as the perception.

We aim at an individualised, decorated payout in consideration of intention, profile, portfolio, context, functionality and storyline progress aspects to be able to solve development troubles. Therefore, we propose a systematic solution of layout development that is based on patterns. Layout patterns as mentioned in [16,17] help to derive possible solutions from an intentional description and allow to reuse components and concepts.

This paper is organised as follows. Section 2 discusses related work and points open issues and needed enhancements. A pattern-based approach is mentioned in section 3 that is able to detect and solve some of the existing development problems. Section 4 proposes dynamic placement to be able to adapt the content and its representation according to application demands and context issues of the website. Finally, we conclude in section 5 and indicate further research directions.

2 Related Work

A lot of HCI approaches [16,11,13,17] have introduced guidelines how to develop good and useful websites, with a range from generic recommendations to directly applicable solutions. For example, Shneiderman [13] has defined eight 'golden' rules of interface design that should be considered for the development of layout. Unfortunately, these general rules are not directly applicable and have to be refined context dependent. According to Herczeg [6], development mistakes can occur in complex situations if the context was not taken into account. For example, in the case of multiple page displays, it is impossible to define a fix and universally valid value, how many changes between two interaction steps are acceptable to achieve an adequate short-term memory load. The acceptable load depends on the possible attention of users, the abilities of the equipment and its application area. Too little changes cause long ways to get to the information, while very short ways result in too many changes. Thus, refinements that consider further aspects of the development are obligatory.

At the opposite end, van Welie [17] has developed a pattern language that tries to benefit from relations between specific layout patterns. Patterns of this language describe problem-oriented solutions based on a specification of well orchestrated layout elements at different levels. They are problem-based, so that this approach results in a huge number of patterns and possible combinations. Unfortunately, it is hard to take all of them into account during the development process and moreover it is hard to handle all valid transitions between the proposed levels. Therefore, it can be useful to classify by application domains

to keep up the decidability. We prefer a split by dimensions (intention, context, story, content, functionality, and presentation) as mentioned in [12] to limit decision problems in the case of multiple domain applications. Van Welie [17] differs in context, application needs, and user needs patterns. These patterns are very useful to create new websites for any field of application, but they do not support changing demands and the progress of websites. Thus, changing demands result in a redevelopment instead of an alternative layout of content that takes these into account.

Moreover, a large number of publications exist (e.g. Itten [7] and Skopec [14]) that discuss the layout from artists and designers view and give some advices how to use and compose graphical elements. These advices help to derive general rules of layout development that are universally valid in consideration of specified context aspects.

3 Placement Prerequisites

A main precondition of placement is the specification of an action space. Flags can help to organise the content within the action space, mostly realised by the use of grids. Frequently, the positioning and representation of content can be based on concepts, so that it is useful to define general rules as patterns. These patterns help to use and combine grids with the aim of easing the whole development process.

3.1 Patterns

We often notice the existence of similar demands in the case of similar applications. Mostly, such constellations result in monotonous representations so that it would be useful to be able to reuse an existing solution as a pattern and adapt it to specific demands of another. This problem-oriented approach is suitable if we have a sufficient list of patterns as built by [17]. In general, patterns help to reuse concepts and components. We are interested in general presentation patterns and proposed to utilise principles taken from cognitive psychology [10]. As a result, we introduced a separation into communication, perception, and composition patterns enhanced by work progress patterns and pattern clusters with the objective of rising chances to reuse concepts while the development process. Patterns can be used to create and derive grids as well as to develop rules how to apply content to the tiles of a grid.

3.2 Grids

Grids were adapted from graphic design and are used for page layout organisation. Usually, a grid splits the whole action space into rectangular tiles that are able to integrate content as mentioned in [8]. In general, a set of grids can be used for partitioning. Further, all grids can consist of a set of subgrids with the objective of increasing their applicability. Grids allow to reuse approved splits to

ease the creation of applications with the same style, indents, sizes and spaces. According to [9], grids are used to be able to place the content in conformity with objective and functional criteria. They define the properties of each tile and the position within the action space.

Typically, application development has to consider global guidelines, e.g., ensuring corporate identity demands. To separate these from situation-dependent definitions, we propose to distinguish between the following types:

- frame grids / tiles
- body grids / tiles

Frame grids and tiles as stable parts of page layout organisation are valid for all pages to ease the orientation within the action space. Therefore, all content object types have to be consistent with these definitions because otherwise some of them cannot be displayed adequately. On the other hand, body grids and tiles concern the representation of non-global content object types. They allow individualisations and depend on the story progress in consideration of frame definitions and restrictions.

3.3 Story

According to [15], stories describe sequences of scenes within applications. They are useful for individualised representations of content and specify possible runs through the application. Stories are needed because users don't act in the same way and their profile and portfolio is very heterogeneous. Typically, several stories are available within a specified story space. It can deal with different demands and allows to adapt the content to the preferences of the users. Stories are important for placement because they often intend a specific progress type that demands for a specific partitioning. If such a partitioning is not compatible with the chosen partitioning of the application, we have to change the application or the story depending on their defined weight.

4 Dynamic Placement

In general, placement can be defined as an assignment of content to a specified action space. A very common placement approach can be realised by positioning the content without any predefined rule. This approach is appropriate to build small and full-custom designed websites, wherefore it is primarily used to create work of art. On the other hand, information-intensive websites specify global grids to ease the positioning by the reuse of placement flags. It allows multiple page displays that are based on a similar representation.

While the first approach causes a huge development effort, the placement by the use of global grids is less flexible, but both of them characterise a static placement of content. We propose dynamic placement that is able to adapt the content and the layout according to application demands and context issues.

4.1 Placement Problems

We call problems *placement problems* if the representation of content is not in line with a chosen layout. Placement problems occur if a layout is more specified than a simple structuring like very common two-column and three-column grids. Particularly, it concerns static placements of information-intensive websites because in these cases a singular placement strategy exists, which is not able to represent all the content conveniently. Simple structuring solutions are easy to realise and can avoid some placement problems but they often result in similar and unemotional representations. Further, they prevent content adaptation in consideration of dimensions as intention, context, and story that were introduced in [12]. While context is important for adaptation of content and layout to different devices and users, the story eases generic specifications of an desired progress and upgrades the orientation of the user within the composition. Moreover, an interplay exists between these dimensions.

To solve placement problems, it is necessary to be able to detect their causes. In the following, we discuss some of the existing placement problems, where the focus is on problems of grids but some attention is given to the content structure, too.

Typical problems of information intensive websites are content over- and underflows. Several methods exist how to solve such problems. It depends on constraints as well as the application domain, which method can be seen as most appropriated.

Content Overflows can be differentiated into the following three types:

- too less space
- too large
- too much content types

Overflows mainly occur if too less space is available for integration of content as a part of a chosen placement tile. In addition, overflows can be a result of placement if content objects are integrable, e.g. by stretching unsized or flexible tiles, but too large regarding the intended presentation impression. To solve such problems, it is necessary to specify acceptable grid tile sizes, aspect ratios, and relations between those. Moreover, overflows occur if web pages consist of a large collection of different content types. We noticed that it is a typical problem of entry pages of portal websites. Such overflows are elusive because they are user-dependent overflows in mind, but in consideration of the user profile splits can help to avoid those.

In general, overflows are solvable by the use of summaries and teasers, which usually have to be defined explicit. Automatic or semi-automatic generated summaries are only possible in some special cases [3]. Another option is the replacement of content by headlines that is useful to create a preview if more than one headline exists. Further, it can be useful to reorganise the content into multiple page displays by a break into sections, chapters, and paragraphs.

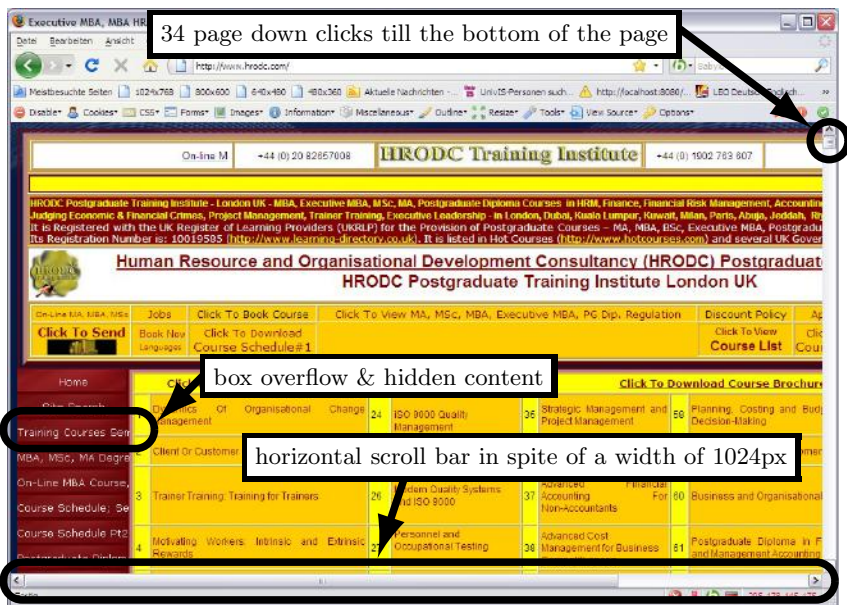


Fig. 1. Content overflow

Figure 1 shows an example of a two-dimensional content overflow. This still existing website overflows horizontal and vertical, although the representation is based on a very common two-column grid. Furthermore, some content cannot be adequately represented because of content box overflows and partly hidden content. Thus, many placement problems exist and a reorganisation is required. We notice that several upgrade strategies are possible and helpful, e.g., headline extraction, breaking into parts, and layout changes. Moreover, layout changes are necessary if menu titles cannot be shortened because of content box overflows and to make hidden content visible.

Content Underflows can be differentiated into the following three types:

- too much space
- too small
- no content type

Content underflows are not so frequently as overflows and addresses nearly the reverse problem. In general, they occur if a content object needs very little space while much space is available within the integrating grid tile. Pages without any content object or useful information are dispensable because headlines within content sections aren't enough. If content objects will be integrated into unsized or flexible tiles, underflows are possible, too, e.g. if the size of a content object is small in comparison to the other objects. An adequate ratio have to be defined application dependent.

The fusion of content objects is an important instrument to avoid underflows, e.g. by fusions of paragraphs, chapters and sections, and aims at an adequate load in consideration of the available tile sizes. Moreover, it is possible to unfold content objects but since we have to consider homogeneity demands we should adapt whole scenes instead of only concerned dialogue steps. Underflow avoidance strategies regarding scenes have to be realised as a pre-check because during the usage adaptations would be applied too late.

4.2 Placement Strategies

Placement strategies concern the placement of content within the action space and within a grid tile. In general, they try to avoid, detect, and solve placement problems with the objective of upgrading the layout. Placement strategies are able to utilise patterns to reuse well-known concepts and components as well as to specify flexible payout rules.

Static placements are widely-used. In general, static placement defines placement tiles, which are able to integrate content. Mostly, unsized or flexible tile specifications are used to avoid subsequent placement problems. Thus, the layout will be adapted to the content. Accordingly, a layout specification has to be adapted each time if new content is available that is inappropriate or not integrable. Otherwise, the representation cannot be ensured. Hence, developers prefer universal solutions and very little attention is paid to the layout. Typically, templates will be used for static placement to be able to change the layout without to change the content. If content shall be assigned to a grid tile, a check is necessary that an embedding is possible. In the case of placement within the action space we further have to consider relations to other tiles, e.g. sizes, distances, and properties. To make this approach more flexible some applications use additional templates that are harmonised with the main template. In the case of changes all of them have to be considered.

A main aim of dynamic placement is to supersede the obligation of manual customisations to be able to adapt the content or layout automatically to changing situations. Applications that are based on stories help to achieve this goal because situation-dependent placement requirements can be defined. Thus, it is possible to create adaptive applications based on these needs and additionally existing rules to solve concurrency problems. In contrast to static placement, dynamic placement is able to adapt the content to the layout and vice versa. Further, it aims at a dynamic adaptation without any manual intervention.

Some work has been done in the field of grid specification and enrichment. Feiner [4] has introduced an approach that is based on grids and tries to automate the display layout completely. Unfortunately, this approach is less flexible regarding customisations to the heterogeneous wishes and demands of the users and applications, e.g. consistency versus alternation. Graf [5] has proposed a constraint-based graphical layout that derives the layout based on topological and geometrical restrictions. This approach is able to adapt the layout to preferences but needs an extensive constraint specification to realise flexibility and avoid wrong options and decisions at the same time.

We notice for nearly all placement problems solutions in reality. Unfortunately, it often is a decision problem in which situation which solution can be seen as most appropriate. However, a systematic solution based on dynamic placement helps to manage the problem.

Therefore, we propose a system as illustrated in figure 2. In a first step, *development dimensions* section gets some meta information from structured content to determine a dimension hierarchy. If user and provider preferences exist, we have to check the provider data against strictness. If customisations are not allowed and provider preferences don't exist, the intention should clarify, which hierarchy is needed based on a set of known applications. If this check has no result, we check the content and its structure against well-known hierarchies. In the case of a result, the result is given to the calculation section. Concurrently, content and its structure was given to the pattern section.

The pattern section decides the initialisation of the calculation process. This process is searching for appropriate patterns within a pattern database. The calculation considers the dimension hierarchy because of existing interactions. Further, meta information is considered to be able to get most appropriate patterns. If a dimension has been found adequate patterns, it gives the result to the next dimension in hierarchy. During the process, a dimension can detect that a presentation with the currently chosen pattern is impossible. Such problems can occur because all dimensions check against partially different properties. In that case, the pattern will be rejected and sent to the previous dimension with a request for recheck. The last dimension gives the result back to the pattern section. This section publishes the result and makes a transfer to the grid section. If no pattern was found that can act a basis for appropriate grids an abort occurs.

Grid section initialises its own calculation section. It tries to derive adequate grids by using preselected patterns in consideration of meta information. Usually, the applicability of grids strongly depends on the context, e.g. application type, user profile, and provider preferences. If this calculation process has no result, grid section asks for a new patterns list as long as necessary. A transfer of grids to the playout section occurs if the grid section was able to build grids for all required content types.

Playout section calculates the real output in dependence of the story, functionality, content, preferences and available grids. Template precalculations increase the performance of this step but in this case prospective placement problems cannot be detected.

4.3 Mashup Placement

Mashups collect content from different sources to recombine these parts within another application. Typically, mashups are more than a selection and composition of content snippets because of possible interactions between these so that the control of the whole application can be enhanced. Mashups are important in the context of placement because typically mashup applications cannot directly influence the source content. Particularly, changes regarding this content have to be handled and can result in presentation errors.

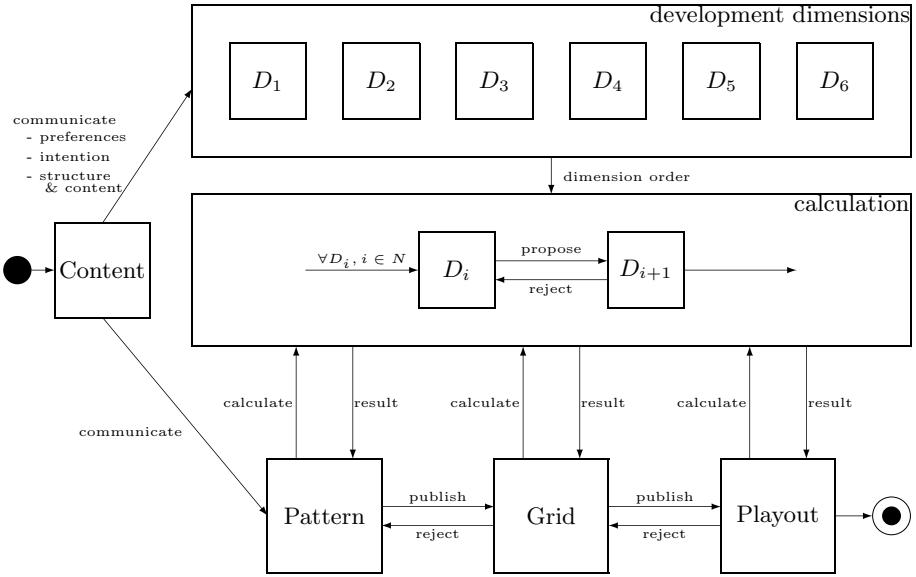


Fig. 2. Dynamic Placement

Some placement and content representation errors can be avoided if either the interface and its functionality is stable or the mashup application becomes robust against changes. Stable means that the interface can be enhanced but not shortened or changed. It is possible to resolve conflicts in a timely manner if only one or a few parts will be collected by the mashup application. In the case of information-intensive websites that collect a multiplicity of external content it becomes problematic.

In general, we are able to distinguish the following four mashup types:

- t_1 - page content
- t_2 - content element
- t_3 - content snippet
- t_4 - content object

All mashup types can contain an interface that is usable by other applications. If this interface will be changed, the stability of the whole mashup application can get lost. In the following, we assume that possibly existing interfaces are stable.

Very common mashup applications collect the whole content from pages including the layout. Such content is hard to adapt to the demands of a mashup page because context information is missing. In addition, the integration can be problematic because of global style and composition definitions that must not be consistent with definitions of the mashup application. Moreover, all updates are able to provoke inconsistencies on either side. The ability to present content is often limited or even impossible if layout changes were made. Another

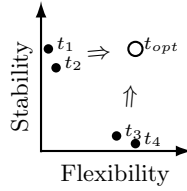


Fig. 3. Mashup types

main problem is that changes against the collected content cannot be detected and processed immediately. Besides partitioning and colouring can change the impression. Mashup type *page content* is inappropriate for content extractions with the objective of controlling other applications. Anyway, this type is stable against changes because all changes are presentable.

If we extract content parts by identifiers, the stability against changes decreases and global aspects can be repressed. In this case, collected content will be represented in the style of the mashup page because global definitions and restrictions of source pages will be neglected. Such *content element* mashups are a bit more flexible because the content is able to change the position within the originally page so that remote layout changes not directly result in mashup problems. If any changes occur against the identifiers, the content cannot be collected. Therefore, the stability decreases a bit.

A third mashup type collects *content snippets* without any style but structure information. It helps to make the content versatile but decreases the stability in dependence on the amount of changes. Changes against the structure that cannot be interpreted by the mashup application immediately result in presentation problems. Thus, this type is very fragile. On the other hand, content accompanied by structure allows to adapt the information to the style of the mashup application. Moreover, individualised representations of the content are realisable with this mashup type.

Finally, *content objects* consist of content snippets that are enhanced by generic style and context information. Style and context information allow to compare the intention of collected content with the demands and intentions of the mashup application. In dependence of existing rules and preferences the content can be adapted. This mashup type is very powerful because of the highest possible flexibility and should therefore result in adequate representations each time. However, the stability is even a bit worse in comparison to the content snippet type.

As illustrated in figure 3, we detected two main mashup dimensions. *Flexibility* describes the degree of adaptability while *stability* concerns the robustness against changes that are not under control of the mashup application. Mashup types, t_1 till t_4 , that are used in figure 3 are consistent with the described types above.

t_{opt} represents an ideal situation that we try to achieve because flexibility and stability are maximised. Thus, it is possible to increase the flexibility of

t_1 and t_2 or the stability of t_3 and t_4 . We propose to take the latter approach because the content already has a structure. Further, the realisation of a stepwise development progress should be easier and is more important in consideration of existing demands in practice.

Several approaches [12] have shown that context-aware mashups are possible that we have characterised as type t_4 . Typically, collected objects have specific notations and their applicability is limited. If we generalise content objects, we can enhance current approaches so that the notations become universal and the applicability is unlimited. Generalised content objects have a meta description to identify, for example, the content type, the intention, and the preferred presentation type. Thus, an enhancement of the term "context" is necessary or we distinguish six dimensions (intention, story, context, content, functionality, presentation) as mentioned in [12] with the objective of upgrading adaptation possibilities.

Current solutions are able to react to changes of remote applications if content objects can be identified. However, mashup page changes cannot be handled adequately at the moment. For instance, style changes can require to choose another representation of content objects or a repartitioning.

Dynamic placement allows to adapt the collected content without any manual intervention with the goal of ensuring adequate representations. The level of adaptation quality depends on available meta information that is assigned to the collected content. At the best, collected content contains meta-data. In the case of absence, the main application can partially fill this gap because it contains the intention of embedding some external content. Problems can occur if no context information is available or can be derived but content over- and underflows are definitely avoidable. In such cases, it is possible to parse the content objects. It is the last and very time-consuming chance to get meta information regarding the content.

5 Conclusion

In this article, we discussed typical placement problems of information-intensive websites as well as their possible solutions. Therefore, we proposed dynamic placement to be able to adapt the content and the layout ad hoc according to existing demands and context issues. Furthermore, we discussed some placement problems and existing strategies to solve these. Afterwards, mashups emphasised the significance of dynamic placement because they consist of elements that are not completely under control of the mashup application. Therefore, we proposed to generalise content-objects of context-aware mashups, because it can avoid manual customisations increase the flexibility.

A subject for future work is the extension of the list of placement problems. For example, specific placement problems occur if we try to place and arrange content within a workspace. In such cases, we need a customisable layout and content as well as functionalities to build sections or collections. Further, we have to refine existing problems. If content elements have internal functionality,

box overflows can occur but could be welcome, for example in the case of alerts. Moreover, we will concretise problems and chances in the mashup field. In that context, a main future challenge is the implementation of an example that shows the problems as well as a possible solution.

References

1. Brodt, A., Nicklas, D., Sathish, S., Mitschang, B.: Context-aware mashups for mobile devices. In: Bailey, J., Maier, D., Schewe, K.-D., Thalheim, B., Wang, X.S. (eds.) WISE 2008. LNCS, vol. 5175, pp. 280–291. Springer, Heidelberg (2008)
2. Daniel, F., Matera, M.: Mashing up context-aware web applications: A component-based development approach. In: Bailey, J., Maier, D., Schewe, K.-D., Thalheim, B., Wang, X.S. (eds.) WISE 2008. LNCS, vol. 5175, pp. 250–263. Springer, Heidelberg (2008)
3. Endres-Niggemeyer, B.: Summarizing Information. Springer, Berlin (1998)
4. Feiner, S.: A grid-based approach to automating display layout. In: Maybury, M.T., Wahlster, W. (eds.) Readings in intelligent user interfaces, pp. 249–255. Morgan Kaufmann Publishers Inc., San Francisco (1998)
5. Graf, W.H.: Constraint-based graphical layout of multimodal presentations. In: Maybury, M.T., Wahlster, W. (eds.) Readings in intelligent user interfaces, pp. 263–285. Morgan Kaufmann Publishers Inc., San Francisco (1998)
6. Herczeg, M.: Interaktionsdesign. Oldenbourg, Munich (2006)
7. Itten, J.: Kunst der Farbe. O. Maier, Ravensburg (1961)
8. Moritz, T., Noack, R., Schewe, K.-D., Thalheim, B.: Intention-driven Screenography. In: Mayr, H.C., Karagiannis, D. (eds.) Proc. of Intl. Conf. on Information Systems Technology and its Applications (ISTA 2007). GI-Edition LNI, vol. 107, pp. 128–139. Köllen Verlag, Bonn (2007)
9. Müller-Brockmann, J.: Grid systems in graphic design. Sulgen/Zurich, Niggli (1996)
10. Noack, R., Thalheim, B.: Patterns for screenography. In: Kaschek, R., Kop, C., Steinberger, C., Fliedl, G. (eds.) International Symposium on Theoretical Programming. LNBIP, vol. 5, pp. 484–495. Springer, Berlin (2008)
11. Norman, D.: The design of everyday things. Basic Books, New York (1988)
12. Schewe, K.-D., Thalheim, B.: Conceptual modelling of web information systems. Data & Knowledge Engineering 54, 147–188 (2005)
13. Shneiderman, B., Plaisant, C.: Designing the User Interface: Strategies for effective human-computer interaction. Addison-Wesley, Boston (2005)
14. Skopec, D.: Digital Layout for the Internet and other Media. Ava Publishing SA, Lausanne (2003)
15. Thalheim, B.: Co-design of structuring, functionality, distribution, and interactivity of large information systems. Tech. Rep. 0315, Cottbus University of Technology, Computer Science Institute (2003)
16. van Duyne, D.K., Landay, J.A., Hong, J.I.: The Design of Sites. Addison-Wesley, Boston (2002)
17. van Welie, M., van der Veer, G.C.: Pattern languages in interaction design. In: Rauterberg, M., Menozzi, M., Wesson, J. (eds.) INTERACT 2003. IOS Press, Amsterdam (2003)

Visual Intelligence Density

Xiaoyan Bai, David White, and David Sundaram

Department of Information Systems and Operations Management
University of Auckland Business School, Auckland, New Zealand
xbai008@aucklanduni.ac.nz, d.white@auckland.ac.nz,
d.sundaram@auckland.ac.nz

Abstract. Advanced visualization systems have been widely adopted by decision makers for dealing with problems involving spatial, temporal and multi-dimensional features. While these systems tend to provide reasonable support for particular paradigms, domains, and data types, they are very weak when it comes to supporting multi-paradigm, multi-domain problems that deal with complex spatio-temporal multi-dimensional data. This has led to visualizations that are context insensitive, data dense, and sparse in intelligence. There is a crucial need for visualizations that capture the essence of the relevant information in limited visual spaces allowing decision makers to take better decisions with less effort and time. To address these problems and issues, we propose a visual decision making process that increases the intelligence density of information provided by visualizations. Furthermore, we propose and implement a framework and architecture to support the above process in a flexible manner that is independent of data, domain, and paradigm.

Keywords: Information Visualization, Intelligence Density, Decision Making, Information Visualization Systems, Framework, Architecture.

1 Introduction

With the rapid advances achieved in the field of information visualization nowadays, various visualization techniques and applications have been widely adopted by decision makers for supporting their decision making activities in all kinds of subject domains such as digital library management, personal information assistants, historical data management, information hierarchy visualization, and so on. Due to the increasingly high complexity involved in the problems that decision makers face, they are seeking visualization systems that could enable them to take better decisions with less effort and time. The complexity is further exacerbated by the fact that most decisions involve multiple paradigms, multiple domains and complex multidimensional data. For better evaluating and comparing the visualization capabilities of various systems, we introduce a new concept called *Visual Intelligence Density (VID)* to measure the amount of useful decision support information that a decision maker gets by viewing visualizations for a certain amount of time (section 4). There are five types of visualization capabilities that are extremely useful for improving VID, i.e. visual contents integration, visualization

solutions comparison, visualization techniques integration, visualization technique customization, and visualization technique creation.

Although existing information visualization technologies and systems may contribute to enhancing VID to some extent, they fail to provide sufficient support for complex problems. To address these problems and issues, we propose a visual decision making process that increases the intelligence density of information provided by visualizations. Furthermore, we propose and implement a framework and architecture of a visualization-oriented decision support system generator (VDSSG) to support the above process in a flexible manner that is independent of data, domain, and paradigm.

This paper is organized as follows. In section 2 visual decision making problems and the requirements of systems to effectively support them are explored. Keeping these requirements in mind, we critique current information visualization systems (IVS) in section 3. The failure of these systems to effectively support decision makers motivates us to define a new measure, namely VID, to evaluate and communicate the visualization capabilities of IVS, in section 4. Section 5 proposes a process to enhance VID. Furthermore, we propose a framework and architecture to support the above process in sections 6 and 7 respectively. We conclude the paper with a scenario-driven description of a prototypical system which realizes the proposed framework and architecture.

2 Visual Decision Making Problems and Requirements

Decision makers nowadays encounter more and more problem issues with increasingly high complexities. These problems often require visualization solutions with higher intelligence density so that more features of the underlying large volume of complex data could be reflected and perceived by decision makers, such as patterns, trends or relationships discovered from the underlying data. In addition, they may occur in various application domains. For example, Card et al. [1] pointed out seven key application domains where these problems may occur, namely, statistical and categorical data management, digital library management, personal services support, complex documents management, history management, classifications management, and networks management. These problems are reviewed and synthesized from four main broad problem categories, namely, data-related problems, paradigm-related problems, visualization-related problems, and user-related problems. The following table briefly summarizes problems under each category.

The problems identified above tend to expose a focal issue of how to support users to visualize, perceive and interpret more useful decision support information within limited visual spaces with less effort and time so as to make better decisions. For

Table 1. A Summary of Visual Decision Making Problems

Problem Category	Problem Description
Data-Related [2, 3, 4, 5]	High data complexity and volume
Paradigm-Related [4, 6]	Multiple data types; Time variant changes & trends
Visualization-Related [7, 2, 8, 6]	Visualization integration, quality, and aesthetics
User-Related [6, 4]	High demands for prior knowledge and training

addressing these problems and issues, five key functional requirements with regard to system implementation are identified and deemed as extremely helpful. They are visual contents integration (i.e. integrating visual contents generated from various visualization techniques to provide users a more comprehensive view of the underlying data), visualization solutions comparison (i.e. allowing users to develop alternative solutions and visualize their outputs simultaneously for comparison purposes), visualization techniques integration (i.e. integrating various visualization techniques within a single system environment), visualization technique customization (i.e. customizing certain visualization technique according to user requirements), and visualization technique creation (i.e. developing new visualizations either from scratch or based on existing reusable visualization components like various transformation or mapping operators). Understanding these requirements may facilitate us to properly review and evaluate the support provided by visualization techniques and systems.

3 Information Visualization Systems

This section is to review, synthesize and criticize the support offered by existing information visualization systems and technologies for addressing the identified problems, issues and requirements. A large number of IVS have been implemented and widely adopted in many application domains for facilitating various visual decision making activities.

Some IVS applications are developed for some specific purposes such as supporting a specific application domain or visualizing a specific data type. A substantial number of specific IVS applications have been reviewed and examined by Chi [9], Chen [10], and Turetken and Sharda [11]. For example, the TileBars system [12] can visualize the result of searching certain terms from a set of documents in the domain of complex documents management, while the Time Tube system [13] enables users to monitor the structure of a website in terms of its hyperlinks and evolvments over time within the domain of networks management. Due to their specific design intentions, this type of IVS often involves single visualization technique and is developed for visualizing certain data type in a limited application domain. Thus, specific IVS by nature tend to provide weak / no support for satisfying the requirements such as visual contents integration, visualization solutions comparison, and new visualization technique creation.

Other IVS applications serve as certain system platforms or environments that allow specific IVS to be generated for addressing structured or ill-structured decision problems. There are quite a lot of IVS generators that have been implemented and some of them still survive in the market, such as Khoros [14] and the spreadsheet for information visualization system [7]. Although these IVS tend to provide good or limited supports for some specific requirements, few of them could provide sufficient and flexible support for all the requirements identified. For example, the spreadsheet system [7] may provide some support for simultaneously viewing the visualizations generated from different data sets and allowing decision makers to compare and operate on the integrated visual contents. However, it provides very little support for creating new visualizations through smoothly integrating existing visualization techniques. In general, existing IVS applications fail to provide sufficient, effective and

comprehensive support for addressing the problems and requirements identified. For better evaluating and communicating the visualization capabilities of the systems which may support these problems, we introduce and define the new term VID in the following section 4.

4 Visual Intelligence Density

This section aims to propound the new term – visual intelligence density and the way of how it is defined. Briefly speaking, the VID definition is derived from the concepts and ideas about intelligence density and visual intelligence. Dhar and Stein [15] introduced a measure called intelligence density to evaluate the power of a decision support technique in terms of providing useful understandable information for supporting decision making activities. Intelligence density can be applied to assess “the amount of useful ‘decision support information’ that a decision maker gets from using the output from some analytic system for a certain amount of time” [15]. Jern [5] defines visual intelligence as a process in which proper information visualization techniques are deployed to aid and enhance decision makers’ capabilities of information discovery and exploitation. The sole purpose of visual intelligence is to amplify decision makers’ cognition activities and decrease the time needed for getting the essence of the underlying data, i.e. “time-to-enlightenment” [5]. By integrating the intelligence density measure into the process of visual intelligence, the new term – visual intelligence density – is introduced to describe and evaluate the intelligence provided by certain visualization technique. VID extends the definition of intelligence density by applying it into the context of information visualization. It will be used in this paper to evaluate and represent the effectiveness and efficiency of a visualization technique. It can also be used to compare the power of similar or different types of visualization techniques. We proposed that *Visual Intelligence Density is a measure of the amount of useful decision support information that a decision maker gets by creating, manipulating, layering and viewing visualization for a certain amount of time*. VID aims to reveal the usefulness and quality of certain visualization technique in terms of supporting decision making activities. This “usefulness” refers to the degree of effectiveness, efficiency and appropriateness that the visualization technique provides for supporting decision making. The concepts and implications of VID may contribute to a deeper understanding of the field of visual decision making which, in turn, inspires us to propose a formal definition of VDM and develop the VDM process.

5 Visual Decision Making Process

VDM concerns the use of visualization technologies or tools for supporting decision making activities. Based on our VID definition and the description of “analytical and visual decision making” [16], we define *visual decision making as activities or processes which deploy visualization technologies or tools to support decision makers to better identify problems, develop better solutions with high or enhanced visual intelligence density, and make better decisions more efficiently and effectively*.

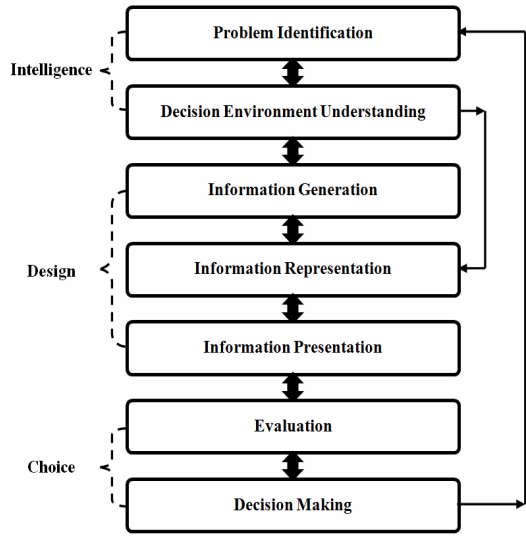


Fig. 1. The Visual Decision Making Process

For supporting VDM, we proposed the VDM process (Fig. 1). This process is composed of several steps that enable VID to be enhanced by generating, transforming, visually encoding and presenting information. In general, the first two VDM stages are dedicated to gaining a good understanding of a decision problem which a decision maker tries to resolve, which correspond to the intelligence phase of the Simon's model [17]. The problem identification stage aims to clearly identify and concisely define the gap between the existing problem situation (AS-IS) and the desired state that the decision maker is trying to achieve (TO-BE). The difference between the AS-IS and TO-BE situations is the actual problem(s) to be addressed. As a result, the problem under investigation could be formally defined so as to formulate a unified understanding of the problem and facilitate the communication of it, for example, defining the boundary and time duration of the problem [18, 19]. Based on the understanding of the problem achieved in the first stage, the second stage requires identifying factors of the decision environment and features of the decision maker / stakeholders that may affect the decision as well as their possible influences. Possible external factors could be those variables that determine the appropriateness of applying the VDM process [20, 21]. Internal factors could be organizational culture and leadership status if the VDM process is to be applied within the organizational environment [19]. These factors or features and their effects on decision making contribute a lot to the later VDM stages in terms of how to design, develop, assess, rank and select visual scenarios.

The following three VDM stages focus on designing and developing alternative visual scenarios, which are consistent with the design phase of the Simon's model [17]. More specifically, in the information generation stage, available source data are transformed or prepared to generate useful decision making data set with higher intelligence density. The information representation stage is responsible for visually encoding the generated useful decision making information. The information presentation stage is to

properly organize the layout of the encoded information so that the decision maker may grasp more useful features of the data with less effort and less amount of time. All these three stages involve the corresponding activities about scenario building, modelling and model management. It deserves to be pointed out that the information generation stage is not compulsory if the available source data can satisfy the requirements for visualization without the need of further transformations.

The last two VDM stages are to select a most proper visual scenario, which are in line with the choice phase of the Simon's model [17]. With alternative visual scenarios being successfully developed, they are assessed and ranked based on a pre-defined set of evaluation criteria in the evaluation stage. Finally, the most proper visual scenario is selected for addressing the defined problem. The whole VDM process could be iterative if further problems or opportunities are detected in the decision making stage. To support this entire VDM process, the VDSSG related concept, framework and architecture are developed and presented in the following two sections.

6 The VDSSG Framework

We define VDSSG as an integrated package of software and hardware that allows decision makers to resolve problems which often involve multiple application domains, paradigms and / or data types by providing flexible support for generating solutions with high visual intelligence density. For realizing a VDSSG application, we proposed the VDSSG framework which consists of four key system components, namely, data, models, solvers, and scenarios (Fig. 2). These components are joined together by a central kernel which is responsible for the communications among different components. Among these components, the models, solvers and scenarios components have further classified sub-types for supporting the three pivotal VDM stages, that is, information generation, information representation, and information presentation.

The VDSSG framework involves two broad categories of data, that is, data provided by users and data required by the VDSSG system executions. The model component can be categorized into three sub types of models, namely, information generation (IG) model, information representation (IR) model, and information presentation (IP) model (Table 2). Such a separation of concerns greatly improves the framework's flexibility and reusability [22]. The solvers follow a similar classification (IG, IR, and IP) and are essentially algorithms that can be applied to manipulate their corresponding type of models. The scenarios can also be divided into three categories. An IG-Scenario is formed by IG-Model, IG-Solver and data provided by users or other IG-Scenarios. It aims to generate the useful decision making information which is often the input of IR-Scenarios. An IR-Scenario could also directly take in the user data if the data provided can present good quality in terms of satisfying the visualization requirements. By integrating proper IR-Model and IR-Solver with the data input, the IR-Scenario could be generated for allowing the data input to be mapped to certain visual objects. The output of the executed IR-Scenario may then serve as the input of an IP-Scenario which also includes certain IP-Model and IP-Solver to define the features and styles of how they look and how they could be interacted by users. The executed IP-Scenario actually produces a visual scenario.

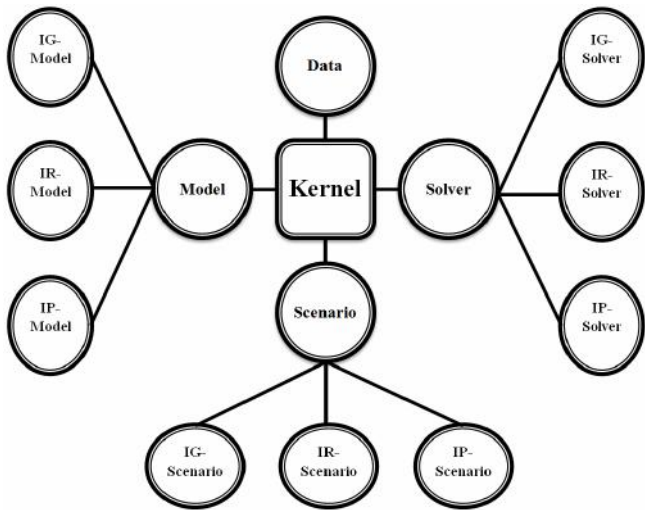


Fig. 2. The VDSSG Framework

Table 2. VDSSG Model Types

Model	Description
IG-Model	Describes and formulates the problem situation that a decision maker is trying to address. It is specifically used for facilitating the generation of useful decision support information.
IR-Model	Describes the problem situation about how the generated decision making information could be transformed into certain visual forms. It is particularly used for supporting the information representation stage.
IP-Model	Describes the problem situation of how visually encoded information could be properly presented so that a decision maker can quickly get the essence of the information and effectively interact with it. It contributes to support the information presentation stage.

All the involved information generation related components, information representation related components and information presentation related components together constitute a visual scenario. Within the visual scenario, each component contained knows nothing about how to communicate with and map to other components. It is the central kernel component that accomplishes the tasks about between-component communications, effectively validating these components and mapping them together. To realize the proposed framework and process, the VDSSG architecture is presented in the following section.

7 The VDSSG Architecture

The VDSSG architecture is developed with integrating good practices of visualization systems design and development (Fig. 3). There are two prominent features about this architecture that deserve to be further discussed. Firstly, the kernel is separated into

four sub components, i.e. information generator, information representer, information presenter, and visual scenario manager. The first three sub components are responsible for the communication and the corresponding components management (e.g. model management and scenario management) in information generation, representation and presentation stages respectively. The visual scenario manager sub component is to generate and manage visual scenarios. Secondly, the mappings among components may occur at two levels, that is, within-scenario mapping and between-scenario mapping. For realizing this architecture and proving the validity of the proposed VDM process and the VDSSG framework, a VDSSG prototype is briefly discussed in the following section 8.

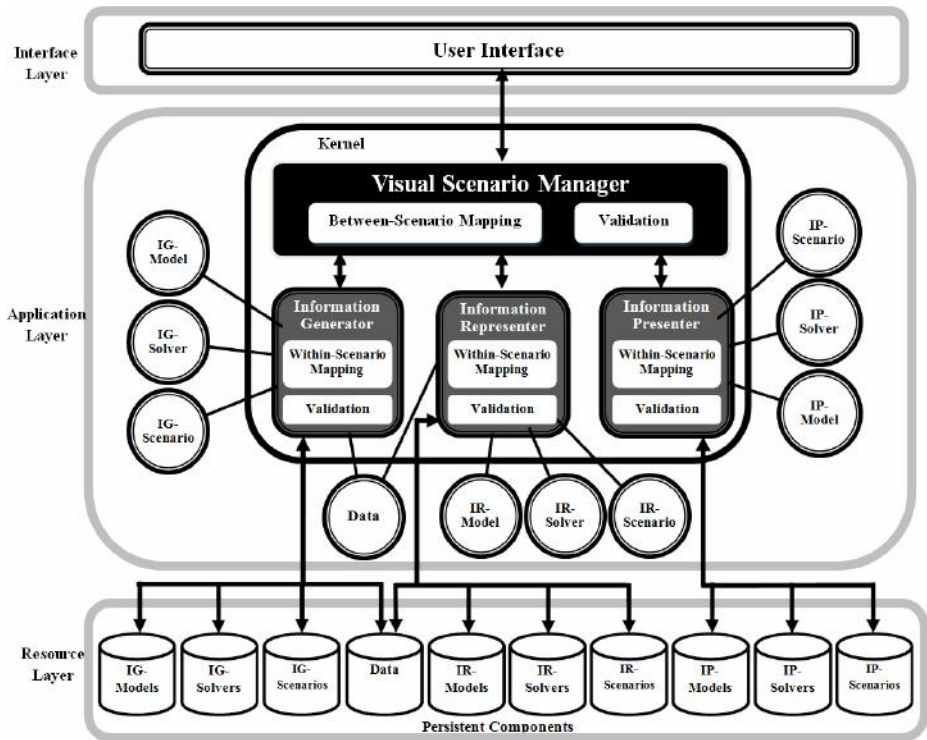


Fig. 3. The VDSSG Architecture

8 Implementation

We implemented a prototypical system to demonstrate the ability and the validity of the VDM process to increase the VID of information regarding a particular problem. The prototype was tested and validated in a number of contexts. In this section, we describe the instantiation of the VDSSG prototype in the context of Napoleon's Russian campaign. This problem is in the nexus of the historical, statistical and

categorical domains where data are complex in the sense that spatial, temporal and multi-dimensional types of data are required to be visualized.

Charles Joseph Minard published a map in 1869 which has been highly recommended by Tufte as “the best statistical graphic ever drawn” [23]. In Minard’s map, there are a number of variables (e.g. the number of survivors, the army’s movements, two-dimensional locations, and temperatures during the army’s retreat) that are plotted and visualized [23].

The prototype focuses on the variables and information relevant to the main force. To make the story even rich, we also extended the case by adding in some make-up data about time (i.e. when the army arrived in / departed from each location on the route) and cooking up some statistical data (i.e. the Russian army allocated at each location, and the required and available resources at each location such as shelter and food).

The prototype is developed by deploying and integrating a series of Microsoft technologies, i.e. Microsoft Windows Presentation Foundation, Microsoft Virtual Earth, Linq to SQL, and ADO.NET. It uses Microsoft SQL Server 2008 to implement the database and Microsoft Visual Studio 2008 as the development platform and Microsoft Windows Vista as the implementation platform.

The implementation of the prototype follows exactly the design of the VDSSG frameworks and architecture and supports the entire VDM process. It is implemented in a flexible and reusable fashion, which enables users to map the same data source to multiple visualizations (i.e. multiple IR-Scenarios) and generate multiple views (i.e. multiple IP-Scenarios) for the same view. Its support for enhancing the VID is illustrated through the following visual scenarios. With the implemented prototype, this so-called “best” map could be made even great.

To support users to explore information about the Napoleon’s Campaign in Russia from 1812 to 1813, visual scenarios are developed by using the implemented prototype. Due to the restriction on the paper length, only three visual scenarios are presented and discussed for proving the increased visual intelligence density.

8.1 A Static Visualization

This visual scenario is quite similar to the original design of Minard’s map in terms of the main force. Only three variables are visualized (Table 3) and the information is presented in a static view (Fig. 4). The width of the black band represents the number of survivors at each location.

Table 3. Features Captured by the Static Scenario

Spatial Features	2-dimensional locations	Temporal Features	None
Multi-dimensional Features	The number of survivors		

8.2 A Simple Animated Visualization

Compared with the previous visual scenario, this scenario shows more information by plotting seven variables (Table 4). Besides features shown in the previous scenario, this scenario also reveals the information about how long the army stayed in each

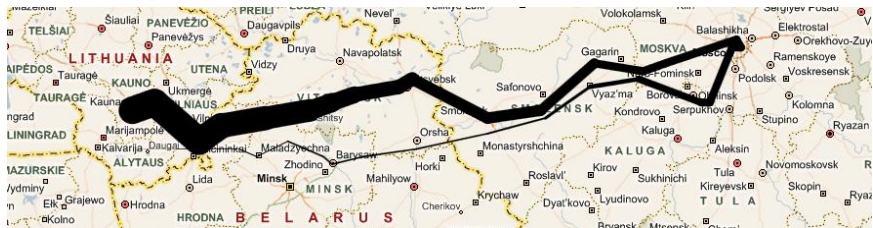


Fig. 4. The Static View of Napoleon’s March in Russia

Table 4. Features Captured by the Simple Animated Scenario

Spatial Features	2-dimensional locations
Temporal Features	When the army arrived and departed
Multi-dimensional Features	Army marching direction and speed, the number of survivors

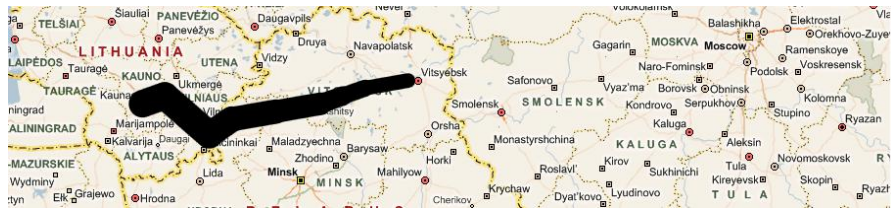


Fig. 5. The Simple Animated View

location, the army moving direction and speed. The information is presented in an animated view (Fig. 5).

8.3 A Sophisticated Animated Visualization

In contrast with the previous two visual scenarios, this scenario shows the most information. Many variables are shown (Table 5) and the information is presented in an animated fashion (Fig.6). Besides the information shown in the visual scenario 2, this scenario deploys bar charts to compare the information of various resources within the same location and among locations. As can be easily detected in this visual scenario, the VID is greatly improved in terms of facilitating users to explore more information of the army march.

Table 5. Features Captured by Sophisticated Animated Scenario

Spatial Features	2-dimensional locations	Temporal Features	When army arrived and departed
Multi-dimensional Features	Army marching direction and speed		
	Number of survivors and temperature at each location		
	Resources available/required at each location such as food and shelter		
	Russian army allocation (e.g. army size and weapons)		



Fig. 6. A Spatio-temporal Animated Visualization with Multi-dimensional features

9 Conclusion

To present visualizations with higher visual intelligence density and better support visual decision making, there are two key requirements which must be satisfied. First, high density decision making data set needs to be generated or prepared based on the underlying source data for visualization. This requirement reveals the need for strong data analysis capabilities. Second, visualizations need to capture more features of the desired decision making data set within the limited visual space so as to present more useful decision making information. However few systems have the ability to simultaneously fulfil both requirements. This paper proposes concepts (VID), process (VDM), framework (VDSSG), and architecture whose explicit purpose is to address these two requirements. Furthermore we have described an implementation of the VDSSG framework and architecture through three scenarios that explore spatio-temporal multidimensional problems and their support.

References

1. Card, S.K., Mackinlay, J.D., Shneiderman, B.: Readings in Information Visualization: Using Vision to Think. Morgan Kaufman Publishers, San Francisco (1999)
2. Hibbard, B.: Top Ten Visualization Problems. ACM SIGGRAPH Computer Graphics 33(2), 21–22 (1999)
3. Santos, S.D., Brodlie, K.: Gaining Understanding of Multivariate and Multidimensional Data through Visualization. Computers and Graphics 28(3), 311–325 (2004)
4. Saffer, J.D., Burnett, V.L., Chen, G., van der Spek, P.: Visual Analytics in the Pharmaceutical Industry. IEEE Computer Graphics and Applications 24(5), 10–15 (2004)
5. Jern, M.: Visual Intelligence – Turning Data into Knowledge. In: Proceedings of IEEE International Conference on Information Visualization, London, pp. 3–8 (1999)
6. Chen, C.: Top 10 Unsolved Information Visualization Problems. IEEE Computer Graphics and Applications 25(4), 12–16 (2005)
7. Chi, E.H., Barry, P., Riedl, J., Konstan, J.: A Spreadsheet Approach to Information Visualization. In: Proceedings of IEEE Symposium on Information Visualization, pp. 17–24 (1997)

8. Keim, D.A., Kriegel, H.P.: Visualization Techniques for Mining Large Databases: a Comparison. *IEEE Transactions on Knowledge and Data Engineering* 8(6), 923–938 (1996)
9. Chi, E.H.: A Taxonomy of Visualization Techniques Using the Data State Reference Model. In: *Proceedings of IEEE Symposium on Information Visualization*, pp. 69–75 (2000)
10. Chen, C.: *Information Visualization: Beyond the Horizon*, 2nd edn., pp. 89–142. Springer, London (2004)
11. Turetken, O., Sharda, R.: Visualization of Web Spaces: State of the Art and Future Directions. *SIGMIS Database* 38(3), 51–81 (2007)
12. Hearst, M.A.: TileBars: Visualization of Term Distribution Information in Full Text Information Access. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 59–66. ACM Press/Addison-Wesley Publishing Co. (1995)
13. Chi, E.H., Pitkow, J., Mackinlay, J., Pirolli, P., Gossweiler, R., Card, S.K.: Visualizing the Evolution of Web Ecologies. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 400–407. ACM Press/Addison-Wesley Publishing Co., New York (1998)
14. Konstantinides, K., Rasure, J.R.: The Khoros Software Development Environment for Image and Signal Processing. *IEEE Transactions on Image Processing* 3(3), 243–252 (1994)
15. Dhar, V., Stein, R.: *Intelligent Decision Support Methods: the Science of Knowledge Work*. Prentice Hall, Upper Saddle River (1997)
16. He, G.G., Kovalerchuk, B., Mroz, T.: Multilevel Analytical and Visual Decision Framework for Imagery Conflation and Registration. In: Kovalerchuk, B., Schwing, J. (eds.) *Visual and Spatial Analysis: Advances in Data Mining Reasoning, and Problem Solving*, pp. 435–472. Springer, Heidelberg (2004)
17. Simon, H.A.: *The New Science of Management Decision*. Harper & Row, New York (1960)
18. Chermack, T.J.: Studying Scenario Planning: Theory, Research Suggestions and Hypotheses. *Technological Forecasting and Social Change* 72(1), 59–73 (2005)
19. Keough, S.M., Shanahan, K.J.: Scenario Planning: Toward a More Complete Model for Practice. *Advances in Developing Human Resources* 10(2), 166–178 (2008)
20. Schoemaker, P.: When and How to Use Scenario Planning: a Heuristic Approach with Illustration. *Journal of Forecasting* 10(6), 549–564 (1991)
21. Chermack, T.: Improving Decision-making with Scenario Planning. *Futures* 36(3), 295–309 (2004)
22. Heer, J., Agrawala, M.: Software Design Patterns for Information Visualization. *IEEE Transactions on Visualization and Computer Graphics* 12(5), 853–860 (2006)
23. Tufte, E.R.: *The Visual Display of Quantitative Information*. Graphics Press, Cheshire (2001)

A Framework for Data Quality in Data Warehousing

Rao R. Nemani¹ and Ramesh Konda²

¹ Opus College of Business
University of St. Thomas
1000 LaSalle Avenue , TMH 455
Minneapolis , MN 55403
Nema8811@stthomas.edu

² Nova Southeastern University
Graduate School of Computer and Information Sciences
3301 College Avenue, Ft. Lauderdale, Florida 33314
konda1991@yahoo.com

Abstract. Despite the rapid growth in development and use of Data Warehousing (DW) systems, the data quality (DQ) aspects are not well defined and understood. Organizations rely on the information extracted from their DW for their day-to-day as well as critical and strategic business operations. Creating and maintaining a high level data quality is one of the key success factors for DW systems. This article examines the current practices of DQ, and proposes a research and experienced-based framework for DQ in DW, which describes an approach for defining, creating, and maintaining DQ management within DW environment.

Keywords: Data Warehousing, Data Quality, Data Warehouse Development Life Cycle.

1 Introduction

Typically, organizations collect data through various systems and try to create coherent and aggregate data for decision-making purpose. Bryan [4] states that fast growth of electronic commerce in the last decade registered a lot of business critical and confidential data being exchanged online among companies and customers. However, this vast amount of stored data failed to provide knowledge because of its raw form. To solve this problem, organizations adopted a transformation process called a data warehouse, which is defined as a “collection of subject-oriented, integrated, non-volatile data that supports the management decision process”. Typically, a data warehouse contains five types of data: current detail data, older detail data, lightly summarized data, highly summarized data, and metadata. Nowadays, data warehousing has become an integral part of both business and Information Technology strategies to enable On-Line Analytic Processing (OLAP) and Decision Support Systems (DSS).

There has been great progress and improvement in core technology of DW; however the DQ aspects are one of the crucial issues that were not adequately addressed.

Ensuring high-level DQ is one of the most expensive and time-consuming tasks to perform in data warehousing projects. Many data warehouse projects have failed halfway through due to poor DQ. This is often because DQ problems do not become apparent until the project is underway. The quality of information systems (IS) is critically important for companies to derive a return on their investments in IS projects, and the DW is no different in that sense.

In this paper, the authors examine the current DQ practices in DW, and propose a framework for improving the quality of data in DW environment. In the Section 2, the problem statement around which this paper offers a solution is discussed. In Section 3, a brief literature review is discussed. Following which in Section 4, a framework for the DQ in DW is presented. Section 5 describes the typical use case. The last section summarizes the paper.

2 Problem Statement

Literature review related to issues with DW reveal that DQ is one of the most prominent issues. Poor data quality increases operational costs, adversely affects customer satisfaction, and has serious ramifications on the society at large. Data quality problems range from a minor insignificant to major problematic issues [3, p. 44].

3 Literature Review

The science of DQ is yet to be advanced to the point where standard measurement methods can be devised for any of these issues. Inappropriate, misunderstood, or ignored DQ has a negative affect on business decisions, performance, and value of data warehouse systems. English [5] argues that managing quality of your information is equally important as managing your business.

In a survey by Friedman, Nelson, and Radcliffe [6], it was stated that 75 percent of survey respondents reported significant problems stemming from defective and fragmented data. There are many sources of 'dirty data', which includes a) poor data entry, b) data missing from database fields, c) lack of company-wide or industry-wide data coding standards, d) multiple databases scattered throughout different departments or organizations, and e) older systems that contain poorly documented or obsolete data [1, p. 309]. Nord [8] mentioned that the DQ has become an increasingly critical concern and it has been rated as a top concern to data consumers in many organizations.

Iain and Don [7] argue that in order to tackle this difficult issue, organizations need both a top-down approach to DQ sponsored by the most senior levels of management and a comprehensive bottom up analysis of data sourcing, usage and content including an assessment of the enterprise's capabilities in terms of data management, relevant tools and people skills. However, the crucial question of defining data quality is often ignored until late in the process. This could be due to the lack of solid methodology to deal with DQ. DQ tools generally fall into one of three categories: auditing, cleansing and migration. Data auditing tools apply predefined business rules against a source database. These tools enhance the accuracy and correctness of the data at the source.

Above discussion indicates that DQ issues are critical and need to be addressed in order to be successful with DW environment.

4 A Framework for Data Quality in Data Warehousing

Xu et al., [11] agree with the notion that DQ means accurate, timely, complete, and consistent data. Vikram and Sreedhar [10] proposed a nine steps approach to for successful deployment of a DQ program for a DW initiative. The nine steps include identifying data elements, defining data-quality measurement, instituting the audit measure, defining target metrics for each data attribute, deploying the monitoring program, finding gaps, automating the cleansing effort, developing procedures and establishing a continuous monitoring program.

Theodoratos and Bouzeghoub [9] had presented a framework with a high level approach that allows checking whether a view selection guaranteeing a data completeness quality goal also satisfies a data currency quality goal. So these authors have used a view to accomplish the data quality requirements in a DW environment.

Data warehousing depends on integrating data quality assurance into all warehousing phases planning, implementation, and maintenance [2, p. 75]. Practitioners in quality control methodology recommend addressing the “root cause” duly considering the following data quality factors:

- 1) Accuracy
- 2) Completeness
- 3) Timeliness
- 4) Integrity
- 5) Consistency
- 6) Conformity
- 7) Record Duplication

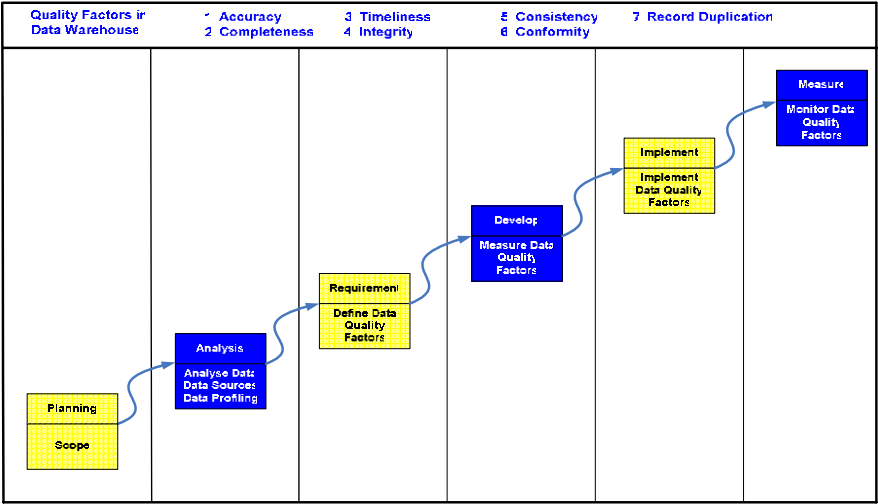


Fig. 1. Data warehouse development life cycle (DWDLC) layers

Based on the literature review and from our personal experience, we propose a comprehensive version of Data Warehouse Development Life Cycle (DWDLC) Layers, which lists comprehensive phases and links the DQ factors as follows.

The major theme in each of the above presented Data Warehouse Development Life Cycle (DWDLC) Layers can be described as follows:

1) Planning: Apart from DQ project success, it is evident that by defining and managing the project scope influences the project's overall success, **2) Analysis:** In this layer, one should consider analyzing the data from various available data sources, hence it is recommended to perform the data profiling of the data, **3) Requirements:** In this layer, DW professional will define and document the required data quality factors for the DW project, **4) Develop:** In this layer, the DW professional will develop and test the DW solution, **5) Implement:** In this layer, the DQ solution will be implemented after duly signed off by the quality assurance team, **6) Measure:** In this layer, a data sampling is done and a measure to understand current process capability is worked out on DQ factors defined in the requirements phase.

Additionally, in order to achieve the above phases, we propose a four-prong DQ management model for defining and ensuring data quality in a DW environment. Under each prong, many relevant tasks need to be defined to and appropriate metrics should be developed to measure the effectiveness of implementation.

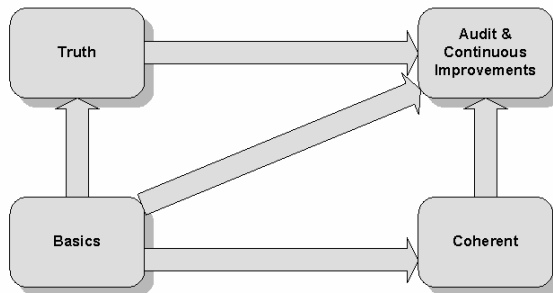


Fig. 2. Data quality in data warehousing four-pronged approach

Each of the prongs in Figure 2 defines specific functional area. The quality in each of these functional areas can be assured via on-going audits and continuous improvement efforts. Each of the prongs is defined as follows in a broader sense.

1) Basics—The very first step in any IT system is to ensure data consistency and completeness. In this case, one may examine individual systems and data sources to ensure the data is complete in the sense that there is no missing data in the fields and has the valid data values. **2) Truth**—Correctness of the data is considered as other side of the coin that shares completeness of the data. One of the major strengths of DW is single point of truth. Using the data from their DW, organizations drive day-to-day as well as long-term strategic business activity. **3) Coherent**—Main premises of DW is merging the data from disparate sources. In this process, it is critical to build the coherent data using dimension keys. Ensuring data coherency is critical for OLAP

analysis as well as building aggregates. **4) Audit and Continuous Improvement**—Plan, Do, Check, and Act (PDCA) process can be used in this stage. As an independent program, a frequent and automated audit of data completeness, accuracy, and coherency will be critical in finding the gaps.

5 Use Case

The following use-case represents some of the scenarios how the above model can be used to address data quality. Consider this scenario: Five functional business managers, each representing a different business function walks into an important business strategic planning meeting. Every one is carrying comprehensive reports about their business functions performance. Each manager is prepared to make some strategic suggestions based on the reports in hand. They have all recognized in less than an hour, their reports reflect entirely different numbers, because the reports are not compiled from a common set of data; no one is sure which, if any, set of numbers are accurate to consider for the strategic planning. This has resulted in postponing the important decision and also initiating crucial initiatives.

The above scenario is a representation of one of the issues faced by organizations across the globe. Using the proposed DWDLC model, the above scenario can be addressed in each layer as follows:

1) Planning: These five functional managers need to plan and scope what reports they are interested to get from the DW, **2) Analysis:** The functional managers should consider analyzing the data from various available data sources and also get the data profiling done, **3) Requirements:** In this layer, these five functional managers will collaborate with the DW professionals to understand the business problem and define and document the required data quality factors for the DW project, **4) Develop:** In this layer, the DW professionals will develop and test the DW solution keeping in mind the DQ factors defined by these five functional, **5) Implement:** In this layer, the DQ solution will be implemented after duly signed off by the quality assurance team, **6) Measure:** In this layer, a data sampling is done and a measure to understand current process capability is worked out on DQ factors defined in the requirements phase. So adhering to the above process will surely minimize, if not totally eliminate the data quality issues.

6 Conclusions

Experience suggests that one solution does not fit all; rather the DQ assessment is an on-going effort that requires awareness of the fundamental principles underlying the development of subjective and objective DQ metrics. In this article, the authors have presented an approach that combines the subjective and objective assessments of DQ, and demonstrated how the approach can be defined effectively in practice.

The goal of any DW and DQ programs is to provide decision makers with clean, consistent, and relevant data. Data Warehouses should provide a “single version of the truth” of high quality data; this enables employees to make informed and better decision while a low quality data has severe effect on organization performance.

A high quality data warehouse increases trust and reliability of data of various applications like data mining and its associated data-reduction techniques. In addition, the trends as identified in the DW can be used to ensure optimal inventory levels, high quality Website design and to detect possible fraudulent behavior. This, in turn, should lead to improved customer satisfaction and an increase in market share.

With the support and commitment from the top-level management and by employing the data quality model and strategy proposed in this paper, the authors confident that an effective data quality can be achieved in a DW environment.

References

1. Andrea, R., Miriam, C.: Invisible Data Quality Issues in a CRM Implementation. *Journal of Database Marketing & Customer Strategy Management* 12(4), 305–314 (2005)
2. Ballou, D., Tayi, G.: Enhancing Data Quality in Data Warehouse Environments. *Communications of the ACM* 42(1), 73–78 (1999)
3. Bielski, L.: Taking Notice of Data Quality: as DQ Discipline goes Enterprise-wide, even the C suite is getting involved. *Banking Journal* 97(12), 41–46 (2005)
4. Bryan, F.: Managing The Quality and Completeness of Customer Data. *Journal of Database Management* 10(2), 139–158 (2002)
5. English, L.P.: Information Quality Management: The Next Frontier. In: *Annual Quality Congress Proceedings*, pp. 529–533. American Society for Quality, Milwaukee (2001)
6. Friedman, Nelson, Radcliffe: CRM Demands Data Cleansing. *Gartner Research* (December 2004)
7. Iain, H., Don, M.: Prioritizing and Deploying Data Quality Improvement Activity. *Journal of Database Marketing & Customer Strategy Management* 12(2), 113 (2005)
8. Nord, G.D.: An Investigation of the Impact of Organization Size on Data Quality Issues. *Journal of Database Management* 16(3), 58–71 (2005)
9. Theodoratos, D., Bouzeghoub, M.: Data Currency Quality Satisfaction in the Design of a Data Warehouse. *International Journal of Cooperative Information Systems* 10(3), 299 (2001)
10. Vikram, R., Sreedhar, S.: Data Quality for Enterprise Risk Management. *Business Intelligence Journal* 11(2), 18–20 (2006)
11. Xu, H., Nord, J.H., Brown, N., Nord, G.D.: Data Quality Issues in Implementing an ERP. *Industrial Management & Data Systems* 102(1), 47–60 (2002)

Visuco: A Tool for Visualizing Software Components Usability

M^a Ángeles Moraga and Coral Calero

Alarcos Research Group, UCLM-INDRA Research and Development Institute
Paseo de la Universidad 4 -13071 -Ciudad Real –Spain
{MariaAngeles.Moraga,Coral.Calero}@uclm.es

Abstract. Component-based technologies are currently beginning to mature and are now widely used in the development of commercial software systems. This has led to the existence of several research works focusing on software component quality. Some of these works propose various measures whose result is normally a number. When dealing with large measures sets, a useful tool through which to facilitate their interpretation is that of visualisation. This paper presents a scalable visual metaphor with which to represent component quality characteristics. Finally, a tool which uses this visual metaphor to show the quality level of usability sub-characteristics has been developed. As an example, the tool has been applied to one real component in order to obtain their usability level.

Keywords: Quality, Visualization, Software Component, Measures.

1 Introduction

Our work combines three main issues: software components, software quality and visualization.

- Software components can be defined as binary units of independent production, acquisition, and deployment that interact to form a functional system [8]. The main process in CBSD (Component-Based Software Development) is that of the component acquisition process which consists of selecting components driven by their quality and their functionality. Nevertheless, the vast majority of efforts have been directed towards the functional aspects of CBSD. However, it is also necessary to take into account the aspects related to quality.
- Software quality is becoming essential to the development of new software. The demand for software quality continues to increase due to our society's increasing dependence on software [7]. However, assessing the quality of software products is not an easy task and requires the definition of a quality model which includes the specific quality characteristics of the software being measured [1].
- Visualization provides an ability to comprehend huge amounts of data [9]. However, representing large data sets implies the use of visual metaphor. A visual metaphor can be defined as the mechanism used to represent information graphically, due to the not inherently physical nature of that information [4].

In order to determine the quality level of a software component, the values of different measures must be assessed. From our point of view, visualization techniques can be used to represent these measures of CSBD. The use of visualization on software components permits a simple and rapid understanding of the component quality features only if the chosen visual metaphor is suitable and well-defined. Bearing all this in mind, in this work we present what we believe to be an adequate visual metaphor with which to correctly visualise the quality of software components. In addition, a tool which supports the visual metaphor is presented.

This paper is structured as follows. Section 2 presents the visual metaphor proposed to represent the software component quality, while in Section 3 the metaphor is adapted to represent the usability quality sub-characteristic. In Section 4 the tool is presented and has been applied to one real component. Finally, conclusions are drawn and future work is outlined in Section 5.

2 Our Proposal

In this section, a general visual metaphor with which to visualize software component quality is proposed. In this metaphor, the software component quality is represented as a building (cube or rectangular prism). Each quality characteristic corresponds with certain attributes of the building. The metaphor is thus able to encode from one to seven quality characteristics.

In order to represent seven characteristics, we propose the use of the following attributes of buildings: height, width, depth, (more or less, depending on bad or good measure results respectively), colour (white to black), orientation (0° to 180°), texture density (large or small bricks, depending on bad or good measure results respectively) and a satellite dish (large or small, depending on bad or good measure results respectively), as is shown in Figure 1.

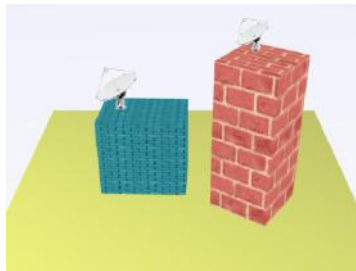


Fig. 1. Example of our extended visual metaphor

3 Adapting the Metaphor to Component Usability

After having defined the visual metaphor, in this section, it will be adapted to assess the usability of a software component. To do this first of all, the measures used to assess component usability must be chosen. In concrete, we decided to commence our work by using the sub-characteristics and the measures proposed in [2], in which the

author adapts the definition of the usability sub-characteristics (e.g. understandability, learnability and operability) proposed in ISO/IEC 9126 [6] to the context of software components. We chose this work because the authors present measures for the usability characteristics that are well-defined and can be assessed automatically. More information about this measures can be found in [2].

The following step is to establish a relationship between the usability sub-characteristics and the visual metaphor. It should be noted that in this case only three elements must be represented: understandability, learnability and operability corresponding to angle, height and colour:

- Angle – Understandability: The angle of the building varies between 0 and 90 degrees. An easily-understandable software component will be represented as 0 degrees and one which is difficult to understand as 90 degrees.
- Height – Learnability: The cube should only grow in one dimension: its height, which varies between 0.5 and 10. A software component which is easy to learn will be represented by a small building, while a software component which is difficult to learn will be represented by a skyscraper.
- Colour – Operability: The colour of the building varies between white and black. A software component which is difficult to work with will be represented in a black texture, while a software component which is easy to work with will be represented in a white colour.

Another important issue to take into consideration is that obtaining a coherent representation depends on normalizing the results of the measures. All the results must, therefore, be standardized between two reference values. In [2] the author define three categories with which to classify the usability of components, according to the IEEE 1061 [5].

Table 1. Threshold values of usability sub-characteristics

Sub-Characteristic	Acceptable	Marginal	Unacceptable
Understandability (CC)	[3-0.95]	(0.95-0.75)	[0.75-(-3)]
Learnability (AC)	[3-1.00]	(1.00-0.80)	[0.80-(-3)]
Operability (OC)	[3-0.90]	(0.90-0.40)	[0.40-(-3)]

In order to obtain a final value of the usability of a component we propose a function which combines the three sub-characteristics (see Figure 2). In this first approach we consider that all the usability sub-characteristics have the same importance in obtaining the consecutive function, although this aspect could be changed by the user according to his/her necessities or priorities.

$$F = \frac{\sum Sub - characteristics\_level}{3}; \text{ where:}$$

Sub-characteristics_level = $\begin{cases} 1 \rightarrow \text{Unacceptable level} \\ 2 \rightarrow \text{Marginal level} \\ 3 \rightarrow \text{Acceptable level} \end{cases}$

Fig. 2. Usability function

4 VISUCO: A Tool to Assess Usability Software Component

In this section, the metaphor previously presented is applied to one software component. To do this, the VISUCO tool which assesses the measure values and represents them using the visual metaphor has been developed. The tool provides two main options: analyze the usability of a software component or see the evolution. The objective of the former option is to show the quality level of the different usability sub-characteristics using the visual metaphor presented in the previous section. In the second case, the tool dynamically shows the differences between two versions of a same software component. However, in this paper we focus on the analyze option.

In order to analyze the software component the user insert the path in which the source code of the software component is located and the tool looks for every code file through the directory hierarchy. It should be noted that the component must be developed using C++. Next, the tool parses these files by calculating the measures needed. Then, it uses the measures previously obtained to compute the usability sub-characteristics. Finally, these sub-characteristic values are presented to the user graphically. The visual representation of the usability sub-characteristics allows the user to discover the usability sub-characteristics of a software component at a glance.

As we have previously mentioned, the tool has been applied to one software component. However, the information related to the component and to the owner company is not revealed for reasons of confidentiality.

The values obtained for the component' usability sub-characteristics are shown in Tables 4. In addition, this table indicates the corresponding attributes in the visual metaphor for each sub-characteristic, along with the measure conversed value (right-hand column of the table).

Table 2. Values of usability sub-characteristics of the first real software component

Sub-Characteristic	Value	Attribute	Normalized Value
Understandability	0.23961231	Angle [0-90°]	41.405820
Learnability	-2.19487328	Height [0.5-10]	8.658122
Operability	-3.64440000	Colour[RGB=0-255,0-255,0-255]	RGB=[255, 0, 0]

These results are visualized by using the proposed visual metaphor: the building. The previous values are converted into visual attributes of the metaphor, by using the normalized values shown in Tables 4 and 5. The visual representation of the software component is presented in Figures 4.

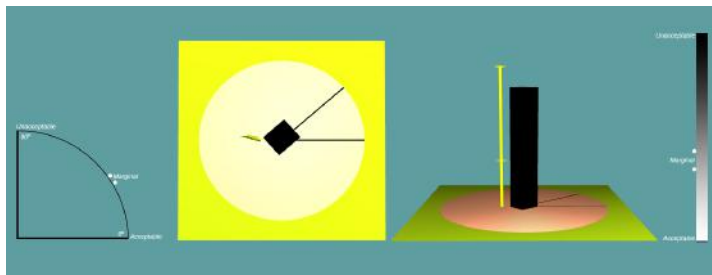


Fig. 4. Visualization of a real software component

As Figure 4 shows, the real software component analysed has an unacceptable usability level. This representation shows a very high black component, corresponding to very bad sub-characteristics of learnability and operability, respectively. However, the angle is of a medium value, and therefore the understandability acceptable.

Obviously, this result coincides with those obtained by applying the function presented in the previous section:

$$F(\text{usability}) = \frac{\text{understandability\_level} + \text{learnability\_level} + \text{operability\_level}}{3}$$

$$\frac{\text{Unacceptable\_level} + \text{Unacceptable\_level} + \text{Unacceptable\_level}}{3} = \frac{1+1+1}{3} = 1 \Rightarrow \text{Unacceptable\_level}$$

However, by using the visual metaphor we have obtained this information without applying the function, and by simply “glancing” at the visual representation.

Another conclusion that can be drawn from the representation is that of which aspects must be improved. For this case, if the usability sub-characteristics of the component need to be improved, then a set of corrective actions must be carried out. With regard to learnability, the quality of the manuals could be improved, thus increasing the explanations of the methods included in the component. In addition, the coupling of the component could be decreased. With regard to operability, it is necessary to improve the quality of the manuals by giving an ample explanation of the configuration options of the component, which should also be increased. Furthermore, the number of methods with return values could be decreased. Finally, understandability could be improved by extending the manuals and decreasing the number of methods with return values.

5 Conclusions and Futures Work

In this paper, a visual metaphor with which to represent software component quality characteristics and sub-characteristics has been proposed. This visual metaphor uses highly familiar elements: buildings and their common attributes, and takes advantage of human visual perception. First, this visual metaphor has been described to represent any quality characteristics and sub-characteristics. It has then been applied to usability quality sub-characteristics, by encoding (or mapping) the three usability sub-characteristics and measures defined in [3]. In addition, the visual metaphor is scalable to encode four more sub-characteristics of any quality characteristic.

Moreover a tool has been developed in order to show the quality level of the usability sub-characteristics and its evolution through two different versions of a same component, in both cases using the visual metaphor. As an example, the tool has been applied to two real software components, obtaining results with regard to the usability sub-characteristics, together with a proposed set of corrective actions with which to improve the usability level.

In our future work we aim to extend the tool in order to visualize the remaining quality characteristics of software components. Moreover, the visual metaphor proposed may be extended to allow the visualization of the usability of large component-based systems that can be also automated.

Acknowledgments. This work is part of the INCOME project (PET2006-0682-01) supported by the Spanish Ministerio de Educación y Ciencia and by the IVISCUS project (PAC08-0024-5991) supported by Consejería de Educación y Ciencia (JCCM).

References

- [1] April, A., Laporte, C.Y.: An Overview of Software Quality Concepts and Management Issues. In: Duggan, E.W., Reichgelt, H. (eds.) *Measuring Information Systems Delivery Quality*, pp. 28–54. Idea Group, USA (2006)
- [2] Bertoa, M.F., Troya, J.M., Vallecillo, A.: Measuring the usability of software components. *Journal of Systems and Software* 79(3), 427–439 (2006)
- [3] Botella, P., Burgués, X., Carvallo, J.P., Franch, X., Pastor, J.A., Quer, C.: Towards a Quality Model for the Selection of ERP Systems. *Component-Based Software Quality*, 225–245 (2003)
- [4] Eick, S.G., Schuster, P., Mockus, A., Graves, T.L., Karr, A.F.: *Visualizing Software Changes*. National Institute of Statistical Sciences (2000)
- [5] IEEE, Std. 1061-1998. *IEEE Standard for Software Quality Metrics Methodology*. IEEE (1998)
- [6] ISO/IEC 9126, *Software Engineering - Product Quality. Part 1* (2001)
- [7] Khan, R.A., Mustafa, K., Ahson, S.I.: *Software Quality. Concepts and Practices*. Alpha Science, Oxford (2006)
- [8] Szyperski, C., Gruntz, C., Murer, S.: *Component Software: Beyond Object-Oriented Programming*, 2nd edn. Addison-Wesley, Reading (2002)
- [9] Ware, C.: *Information Visualization*. Morgan Kaufmann, San Francisco (2000)

Study of Using the Meta-model Based Meta-design Paradigm for Developing and Maintaining Web Applications

Buddhima De Silva and Athula Ginige

University of Western Sydney, Locked Bag 1797, Penrith South DC, 1719, NSW, Australia
bdesilva@scm.uws.edu.au, a.ginige@uws.edu.au

Abstract. When an information system is introduced to an organisation it changes the original business environment thus changes the original requirements. This can lead to the changes to the business processes that information system supports. Also user requests for more functionality as they get used to the system. This gives rise to a cycle of changes known as co-evolution. One way to facilitate co-evolution is to empower end-users to make changes to the web application to accommodate the required changes while using that Web application. This can be achieved through meta-design paradigm. A fundamental objective of a meta-design paradigm is to create socio-technical environment that empowers users to engage actively in the continuous development of systems rather than be restricted to the use of existing systems. Meta-design paradigm can be realised: 1) by providing a technical infrastructure to develop open ended systems that allow end-user developers to evolve the systems; 2) by creating a learning environment which supports end-users to learn and involve in activities required to create / maintain Web applications; and 3) by building a socio-technical environment that allows end-users and professional developers to collaborate in development and maintenance of the system. We developed the necessary technical infrastructure to support meta-design paradigm based on a meta-model of Web applications. This Meta-model based meta-design paradigm supports the development of different types of Web applications required by business organisations and was implemented using Components based eApplication development and deployment System (CBEADS). Using this system we studied how meta-model based meta-design paradigm can be used to develop web applications for three Small to Medium Enterprises (SMEs). This study shows that the meta-model based infrastructure can help to establish the infrastructure, learning environment and socio-economic environment to empower end-users to develop Web applications without restricting them to be passive users of the systems.

Keywords: Meta-design paradigm, end-user development, Web application development.

1 Introduction

The characteristics of the web such as ubiquity and simplicity make it a suitable platform to disseminate information and automate business processes. AeIMS research

group at University of Western Sydney has been working with businesses in Western Sydney region to investigate how Information and Communication Technologies (ICT) can be used to enhance their business processes[1-3]. In this work, we have identified that once an information system is introduced to an organisation it changes the original business environment thus changing the original requirements. This in turn will change the processes that are supported by the information system. Also user requests for more functionality as they get used to the system. This gives rise to a cycle of changes known as co-evolution [4]. One solution to overcome this situation is to empower end-users to develop and maintain the information systems. This can be achieved through meta-design paradigm.

Meta-design paradigm should provide the objectives, techniques, and processes for creating new media and environments allowing ‘owners of problems’ (that is, end-users) to act as designers [5, 6]. In different domains such as information technology, digital networks and nanotechnologies, people have used meta-design paradigm to empower end-users to act as designers rather than limiting them to be consumers [7]. Researchers have used an appropriate technical infrastructure, a set of meta-design tools, and a socio technical environment to empower end-users to evolve the applications in different fields [7-9].

We have developed a meta-model based meta-design paradigm to develop different types of Web applications [10-12]. These different types of Web applications include information-centric Web applications (where the focus is on effective presentation of information), data-intensive Web applications (where the focus is on efficient presentation and management of structured data such as product catalogue) and Workflow-intensive Web applications (where the focus is on efficient automation of business process such as an order processing system). In this paper we report our findings on using the meta-model based meta-design paradigm to develop and maintain information-centric Web applications by three Small to medium Enterprises (SMEs). This paper answers the research question: How to use the Meta-model based Meta-design paradigm to support business organisations to develop and maintain Web applications. First the Web applications were developed and deployed. Then we studied the evolution of the applications during first three months. The findings show that the successful combination of infrastructure, learning environment, and socio-economic infrastructure can help to empower end-users to evolve their applications. Section 2 presents the meta-model based meta-design paradigm to develop Web applications. In section 3, we present the case study on using meta-model based meta-design paradigm to develop information-centric Web applications. Section 4 reviews the related work on meta-design paradigm. Section 5 concludes the paper.

2 Related Work

The existing end-user development tools focused on Web applications have been developed targeting a specific application category. Most of the tools support only data-intensive Web applications. Denim [18] supports information-centric Web applications. Therefore, if an end-user wants to develop different types of Web applications, then he needs to learn to use different end-user development tools and environments. The OOWS [19] based EUD tool is an exception to the other existing end-user tools

since it can support different types of Web applications. However, for the success of that approach, first, the type of ontology needed to support different types of Web applications has to be completed. Then, the tool needs to be evaluated.

The existing commercial Web application development tools such as Microsoft FrontPage and Macromedia Dreamweaver also provide easy to use interfaces. However, the experience with SMEs shows that usability of a tool is not sufficient for the success of end-user development since the learning environment and the socio-technical environment are also important factors for the success of the end-user development.

3 Meta-model Based Meta-design Paradigm to Develop Web Applications

Meta-model based meta-design paradigm should focus on providing a technical infrastructure to develop an open ended system that allows end-user developers to evolve the systems, a learning environment which supports end-users to learn and involve in activities required to create and / or maintain Web applications, and a socio-technical environment that allows end-users and professional developers to collaborate in development and maintenance of the systems. This section explains how the meta-model based meta-design paradigm provides the infrastructure, learning environment and socio-technical environment to support end-user development of Web applications.

3.1 Infrastructure

We first developed a Meta-model for Web applications by analysing patterns of different types of Web applications. The different types of business web applications can broadly be categorised into three groups called information centric Web applications (simple web sites with unstructured information. The focus is on effective presentation of information), Data intensive Web applications (the focus is on efficient presentation on structured data such as product catalogue.), and Workflow-intensive Web applications (the focus is on efficient automation of business processes consisting of sequence of steps such as order processing system.) [13]. Based on the meta-model of web applications, we created an open system which can be designed and modified by end-user developers according to their evolving requirements after deployment of the system. It provides tools for creating Web applications allowing owners of the problem (that is end-users) to act as designers with the help of professional developers.

A web application has many aspects such as “overall layout and look and feel”, “primary navigation”, and “page composition”. Our meta-model provides a way to express these aspects by specifying the attributes associated with these aspects at the conceptual level [13]. . For example, the meta-model of workflow intensive Web applications can be instantiated to create a leave application system where the employees can apply for leave or an order processing system where customers can order items by specifying the attributes of the specific application. Web applications then can be modified by changing the values for the attributes of the meta-model. Meta-model of Web applications is organised into three abstract levels: Shell, Applications and Functions as shown in Fig. 1. [12].

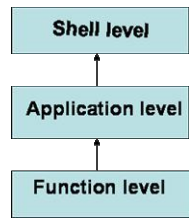


Fig. 1. Three level Meta-model

The aspects common to all Web applications such as navigation and access control are modelled at shell level. The aspects common to specific Web applications such as workflows are modelled at application level. The requirements specific to a view or user interface required to perform actions in an application are modelled at the function level. The meta-model presented in this thesis is implemented by extending the Component based eApplication Development/Deployment Shell (CBEADS) [11].

CBEADS framework consists of engines that can instantiate the Web applications from appropriate specifications. It has tools that can be used to specify the attributes of the meta-model [14]. CBEADS can support the creation of different types of Web applications mentioned before. CBEADS also consists of a function called “create function” which allows the systems to add more functionality to an application [15]. It has a built-in user management application which can be used to change the attributes of the aspects such as user, group, primary navigation, and access control. Further CBEADS provide a run time programming environment that allows users to switch between development and runtime environment with a mouse click.

Using tools available in CBEADS we developed a set of meta-design tools for end-users to create or modify the Web applications [11]. These end-user development tools should design for ease of use and to match the different levels of skills and knowledge of end-user developers [16]. The tools to manage Meta-model provide the easy to use interfaces which include WYSIWYG editors, and drop down menus. The activities required to specify the attribute values in the meta-model to create or modify the Web applications have different levels of complexities. We have categorised these activities in to three complexity levels: 1) Routine level; 2) Logical level; and 3) Programming level [10]. Routine level consists of activities where user can feed direct data from problem domain without much manipulation. Logical level consists of consists of activities that require end-users to formulate the problem domain concepts in a given format and derive data required to populate the meta-model attributes. Programming level consists of activities that require users to code the aspects.

3.2 Learning Environment

In a meta-design paradigm a learning environment should be provided to allow gradual transition of end-users from passive consumers to users, and power users [8]. In other words, the end-user development environment should be a gentle slope system that provides a gradual learning environment for end-users.

The different complexity levels of the meta-model activities help end-users to participate in Web application development at different levels based on their knowledge

and skills. For example, they can reuse a template available in the framework and concentrate on creating the content for the Web site. Then if the end-user developers want to perform advanced development activities, they can learn and customise the templates of the Web site using a form based interface. If they want to create their own template they can write the template files, CSS style sheets using an editor. The complexity of these three activities ranges from routine level to programming level complexities. However, even if end-users can not advance to activities with the complexity of the programming level they have the opportunity to achieve the desired appearance of the systems as explained in next section.

3.3 Socio-technical Environment

A meta-design paradigm should create an environment to support users to collaborate and participate in development of the Web applications. As discussed in previous sections the infrastructure and learning environment created with the meta-model based meta-design paradigm provide end-user developers with an opportunity to actively engage in development of Web applications. However, they are not limited by their capabilities or skills or other factors from using the meta-design paradigm to get the application developed according to their requirements. Meta-model supports the users with different levels of capabilities to participate in the development activities.

The properties- inheritance and overriding of the meta-model can solve the issues with different levels of capabilities of end-users [10]. For example, end-user developers may not have the expert knowledge to specify the template of the Web site. However, if the end-user is provided with a template, he may be able to specify the content of the site. In this situation they can use the templates provided at the shell level. The whole application inherits the presentation properties specified at the shell level. In some cases users may want to develop specific templates for their Web sites. In that case they can override the default template and use the presentation styles provided at the shell level. If the end-users can not perform this activity themselves they can get help from professional developers. The properties of the meta-model together with the complexity levels of activities support collaborative development according to the requirements of the end-users. For example, they can reuse a template available in the framework and concentrate on creating the content for the Web site. If they want a different look and feel, but do not have the required skills to accomplish it, then the meta-model will help end-users to identify the different activities where they can get the help of professional developers. Such a socio-technical environment can eventually lead to end-users being able to develop complete applications. Next section presents a case study on using the meta-model based meta-design paradigm.

4 Case-Study

We used meta-model based meta-design paradigm to help three SMEs to develop and maintain Information centric Web applications. This study was a part of a bigger eCollaboration project which was organised by Austool Ltd at Ingleburn[17]. A group of three SMEs in the toolmaking industry, with limited ICT experience and no Web sites participated in this study. The overall aim of the research project was to conduct

a pilot study to investigate how the toolmaking industry in Australia can benefit from advances in information and communication technology to become globally competitive. The specific aim of this research study was to provide SMEs with a basic Web site and a tool to maintain the Web applications.

First a SWOT analysis (Analysis of Strength, Weakness, Opportunity and Threat) of the SMEs was conducted. This helps to identify the factors that can affect the project as follows.

- ICT infrastructure: ICT was used only to support manual tasks and operations. One was using a computer to manage the accounts.
- Existing IT support: None of the toolmakers has in-house IT support staff. Two had help from their children.
- Time: Toolmakers were working in their business while doing the office administration and marketing. Sometimes they get help from family.
- Affordability: Toolmakers were struggling in their businesses at the time so they wanted to find a solution through collaboration.
- Capabilities of SMEs related to software applications: All three tool makers did not have formal education in IT. However, they were familiar with e-mail, word, excel, CAD CAM. None has developed Web sites.

The impact of these factors on the project was analysed during the feasibility study. This analysis helps to specify the guidelines to the meta-design approach as given in Table 1. Since the tool makers were busy with day to day business and they had no experience in Web application development, it was recommended that design and development of the Web applications was done by the AeIMS research group. The initial Web site is equivalent to the 'seed' in the meta-design process model explained by Fisher, G. [5]. Then the SMEs were provided with the CMS tool within a CBEADS framework to manage the routine level and logical level activities related to the maintenance of their Web site. In other words the users of the system were provided with a tool to evolve the 'seed'.

Three student groups enrolled in the eBusiness course undertook the development of three Web sites. It was planned to gather the requirements for the Websites from SMEs. Meetings were scheduled early morning or at toolmaker's convenience so their

Table 1. The guidelines to meta-design approach

Factor	Recommendations to suit the impact
ICT	Need to get a computer which deployed the CBEADS framework and tools.
IT support	Need IT support for some time until SMEs are competent in managing their applications.
Time	Development phase: uninterrupted during the design phase. Maintenance Phase: Need to make the user interfaces of the tool closer to the user interfaces that SMEs are familiar with.
Affordability	Get the student groups to develop the applications so it provides a win-win situation for both toolmakers and students.
Capabilities	Need to provide tools to similar to the tools that they use now.

business activities were uninterrupted. The meetings between the project team and the SMEs were organised by researchers in the AeIMS research group. Members of the research team made presentations in the meeting. This guaranteed that always the required data and design decisions were made for the betterment of the toolmakers. For example, they facilitated the meeting by showing previous Web sites developed for SMEs by the group. The researchers had to organise several design meetings to create and finalise the look and feel and structure of the Web sites. However, there were delays in specifying the requirements from SMEs. Students completed the Web sites, by the deadlines with the decided structure, look and feel and the made up content using CMS tool in CBEADS. It took longer than expected to complete the Web sites. Researchers had to help SMEs with the content and images for the Web site. After they got the initial Website developed, SMEs took the initiative to evolve the seed Website by coming up with modifications to the structure, look and feel and content.

CMS tool in CBEADS is designed to populate the meta-model instances of information centric Web applications shown in Fig 2. The information-centric Web applications have the common pattern “create-manage-publish information”. In these web applications users need to create the Web pages, manage the Web pages and publish information. The meta-model of information-centric web applications at conceptual level are site structure, Web pages and content.

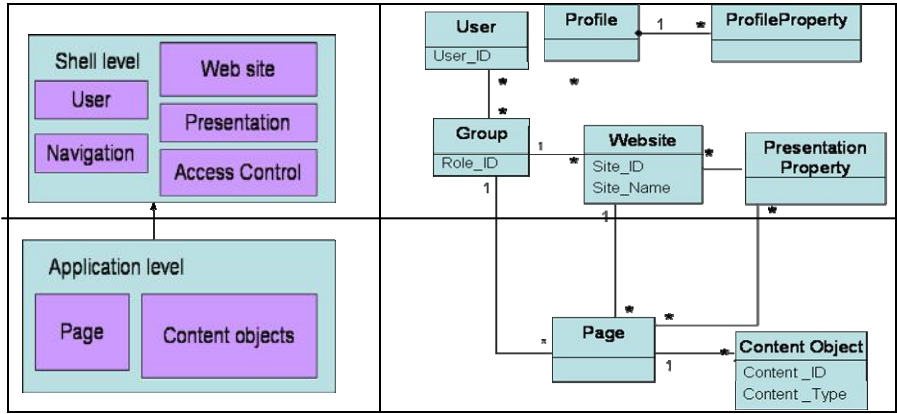


Fig. 2. Meta-model for Information centric Web applications

The hierarchical meta-model for information-centric Web applications consists of shell level and application level. Common aspects required in most of the information-centric Web applications are identified by analysing many information centric Web applications at the conceptual level. These common aspects include user model, access control model, navigation model and meta-Web site object model. These models are stored at shell level. Each user has properties such as name, address, and e-mail address. Each user belongs to one or many groups. The group called “web” can access the functions of the CMS application. The users of the “web” group can manage Web sites. The functions of the CMS applications are associated with a menu link called “CMS”. Authorised users could access the functions of content management

system to create site, manage content and publish it. The information related to the site such as name, owners user_id and the hosting details are stored at the shell level in meta-Web site object model. A Web site consists of many pages. Therefore, the aspects required to model a Web site are stored at the application level. Pages inherit the presentation properties from the shell level. But, it can override at the page level. Page content and page level access control data are stored at this level.

CMS tool is implemented using end-user development technologies such as WYSIWYG editor, minimalist approach, and dropdown menus. WYSIWYG (What You See Is What You Get) editor provides MS word like interface to manage the content of the Websites. WYSIWYG text editor converts the text to the HTML code. Users can insert images by selecting the image from a list of available images. This is similar to browse and attach documents in e-mail applications. Minimalism approach was used as a guideline to design the instruction in user interfaces of CMS tool. Dropdown menus allows end-users to see the existing sites/ and pages when they attempt to manage a site, therefore reduce the amount of information they have to keep in memory. An example user interface of CMS tool is given in Fig 3. Toolmakers were provided with the user training along with a manual.

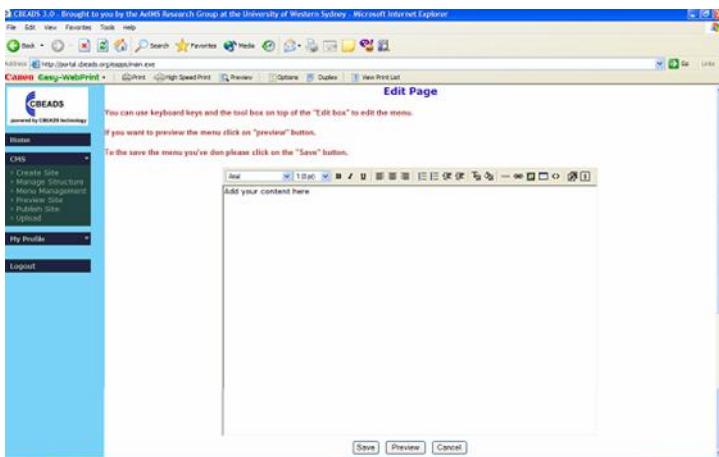


Fig. 3. WYSIWYG Editor in CMS applications. User Interface also gives minimum instructions to use the editor.

The set of functions required to specify and / or modify the attributes of different aspects in information centric Web applications using CBEADS tools along with their level of complexity are given in Table 2.

SMEs feedback on the easiness of the maintenance tasks is given in Table 3. All these tasks introduced to them are at the routine level of complexity. The feedback rating is explained as: 1: I can't do it, it's too difficult; 2: I need some assistance at first, but can do it on my own after; 3: I can do it after explanation and referring to the manual; 4: I can do it in myself by referring to the manual. The feedback from the SMEs shows that they were confident that they can continue doing all the tasks by following the manual or with some assistance for the first time.

Table 2. Functions to manage information-centric Web applications

Aspect	Functions	Complexity Level
Presentation	Select template and style.	Routine Level
	Design the template and styles using forms and visual tools.	Logical Level
Content	Code the styles, templates.	Programming Level
	Add/Modify content in a page.	Logical Level
	Publish content.	Routine Level
Primary Navigation	Change the order in menu.	Routine Level
Access Control	Specify the menu using a form.	Logical Level
	Add User.	Routine Level
	Update User Profile.	Routine Level
	Assign users to roles.	Logical Level
	Assign functions to groups.	Logical Level

Table 3. Feedback on development activities related to different aspects with routine level complexity

Aspect	Toolmaker 1	Toolmaker 2	Toolmaker 3
Presentation	4	3	3
Content	4	3	3
Primary Navigation	3	3	3
Access Control	3	3	3

The maintenance activities (i.e. the evolution of the seed Web sites) carried out during the first three months are listed Table 4. It shows that SMEs have been involved in activities at the routine level. When the required activity is complex the end-users required support from the developers.

Table 4. Maintenance Activities

Activity	Complexity of task	Who is responsible
Updating the employment oppor- tunities	Routine level	SME
Updating the product data	Routine level	SME/ Researcher
Change of banner	Programming level	Researcher

4.1 Discussion

The infrastructure for this project was implemented in CBEADS based on the meta-model of information centric Web applications. The meta-model of information centric Web applications helped the users to change the attributes of the aspects at the conceptual level to evolve the applications according to changing requirements. The experience with toolmakers during the requirement gathering and design phase shows that it would be difficult to get the application developed without significant help from the researchers. This was mostly due to the time limitations and enormity of the

task. Then the observations during the maintenance and the feedback from the tool-makers show that SMEs could evolve their Web site with initial support from the research group. This verifies that the meta-model based development approach had provided a suitable learning environment for the SMEs. Establishment of the complexity levels of the activities identifies the activities in which end-users can participate. Therefore, the meta-design paradigm can help to achieve a balance between the DIY professional developers. This helped to create a socio-technical environment of collaboration between users and developers. Reflection on the experience of the project leads to identification of the factors that affect the implementation of the meta-design paradigm. These factors include characteristics of individuals, support, and infrastructure. These factors are described in Table 5.

Table 5. The factors affecting the meta-design approach

Factor	Observation
Characteristics of Individual	The individuals should be self motivated and keen to learn new technologies.
Support	More support should be provided during the development phase to get the application developed according to the requirements. Then end-users are able to manage most of the activities during the maintenance phase.
Infrastructure	Management of infrastructure should be outsourced to reduce the barrier to end-user development.

5 Conclusion

This paper presents the meta-model based meta-design paradigm to develop Web applications. The case study on meta-model based meta-design paradigm to develop information centric Web applications shows how to develop the infrastructure, learning environment and the socio-technical environment to support organisations to develop Web applications. In this study with three SMEs, we find that the meta-design approach can be used to support co-evolution of Web applications. The case study shows that the end-users were able to participate in activities with routine level complexity. This shows that meta-model based meta-design paradigm helps to empower end-users to develop Web applications without restricting them to be passive users.

References

1. Arunatileka, S., Ginige, A.: Applying Seven E's in eTransformation to Manufacturing Sector. In: eChallenges (2004)
2. Ginige, A.: From eTransformation to eCollaboration: Issues and Solutions. In: 2nd International Conference on Information Management and Business (IMB 2006), Sydney, Australia (2006)
3. Ginige, J.A., De Silva, B., Ginige, A.: Towards End User Development of Web Applications for SMEs Using a Component Based Approach. In: Lowe, D.G., Gaedke, M. (eds.) ICWE 2005. LNCS, vol. 3579, pp. 489–499. Springer, Heidelberg (2005)

4. Costabile, M., Fogli, F.D., Marcante, A.: Supporting Interaction and Co-evolution of Users and Systems. In: *Advanced Visual Interfaces, AVI* (2006)
5. Fischer, G., Giaccardi, E.: A framework for the future of end user development, in *End User Development: Empowering People to flexibly Employ Advanced Information and Communication Technology*. In: Lieberman, H., Paterno, F., Wulf, V. (eds.) *A framework for the future of end user development, in End User Development: Empowering People to flexibly Employ Advanced Information and Communication Technology*. Kluwer Academic Publishers, Dordrecht (2004)
6. Fischer, G., Scharff, E.: Meta-design: design for designers. In: Fischer, G., Scharff, E. (eds.) *Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques*. ACM, New York (2000)
7. Giaccardi, E.: Principles of Meta-Design. In: *School of Computing*, p. 437. University of Plymouth, Plymouth (2003)
8. Fischer, G., et al.: Meta Design: A Manifesto for End -User Development. *Communications of the ACM* 47(9), 33–37 (2004)
9. Giaccardi, E.: Meta-design as an emergent design culture. *Leonardo* 38(4), 342–349 (2004)
10. De Silva, B., Ginige, A.: Solving Design Issues in Web Meta-Model approach to support End-user Development. In: *International Conference in Software Engineering and Data Technologies (ICSOFTE 2007)*, Barcelona, Spain (2007)
11. Ginige, A., De Silva, B.: CBEADS: A framework to support Meta-Design Paradigm. In: Stephanidis, C. (ed.) *HCI 2007. LNCS, vol. 4554*, pp. 107–116. Springer, Heidelberg (2007)
12. De Silva, B., Ginige, A.: Meta-Model to support End-user Development of Web based Business Information Systems. In: Baresi, L., Fraternali, P., Houben, G.-J. (eds.) *ICWE 2007. LNCS, vol. 4607*, pp. 248–253. Springer, Heidelberg (2007)
13. De Silva, B., Ginige, A.: Modeling Web Information Systems for Co-Evolution. In: *International Conference in Software Engineering and Data Technologies (ICSOFTE 2007)*, Barcelona, Spain (2007)
14. Ginige, A., et al.: Smart Tools to support Meta-Design Paradigm for Developing Web based Business Applications. In: *International Conference in Web Engineering*, Como, Italy (2007)
15. Ginige, A.: Re Engineering Software Development Process for eBusiness Application Development. In: *Software Engineering and Knowledge Engineering Conference -SEKE 2002*, San Francisco, Bay, USA (2002)
16. Rode, J.: Web Application Development by Nonprogrammers: User-Centered Design of an End-User Web Development Tool. In: *Computer Science*, p. 292. Virginia Polytechnic Institute and State University (2005)
17. Arunatileka, S., et al.: A Pilot Project on eCollaboration in the Australian Toolmaking Industry. In: *18th Bled eConference eIntegration in Action International Conference (BLED)*. AIS Electronic Library (AISeL), Bled (2005)
18. Newman, M.J., et al.: An Informal Web Site Design Tool Inspired by Observation of Practice. *Human Computer Interaction* 18, 259–324 (2003)
19. Valderas, P., Pelechano, V., Pastor, O.: Towards an End-User Development Approach for Web Engineering Methods. In: Dubois, E., Pohl, K. (eds.) *CAISE 2006. LNCS, vol. 4001*, pp. 528–543. Springer, Heidelberg (2006)

Lost in Translation? Transformation Nets to the Rescue!

Manuel Wimmer¹, Angelika Kusel², Thomas Reiter²,
Werner Retschitzegger³, Wieland Schwinger², and Gerti Kappel¹

¹ Vienna University of Technology, Austria

`lastname@big.tuwien.ac.at`

² Johannes Kepler University, Austria

`lastname@ifs.uni-linz.ac.at`

³ University of Vienna, Austria

`werner.retschitzegger@univie.ac.at`

Abstract. The vision of Model-Driven Engineering places models as first-class artifacts throughout the software lifecycle. An essential prerequisite is the availability of proper transformation languages allowing not only code generation but also augmentation, migration or translation of models themselves. Current approaches, however, lack convenient facilities for debugging and ensuring the understanding of the transformation process. To tackle these problems, we propose a novel formalism for the development of model transformations which is based on Colored Petri Nets. This allows first, for an explicit, process-oriented execution model of a transformation, thereby overcoming the impedance mismatch between the specification and execution of model transformations, being the prerequisite for convenient debugging. Second, by providing a homogenous representation of all artifacts involved in a transformation, including metamodels, models and the actual transformation logic itself, understandability of model transformations is enhanced.

Keywords: Model transformation, runtime model, Colored Petri Nets.

1 Introduction

Model-Driven Engineering (MDE) places models as first-class artifacts throughout the software lifecycle [3]. The main promise of MDE is to raise the level of abstraction from technology and platform-specific concepts to platform-independent and computation-independent modelling. To fulfil this promise, the availability of appropriate *model transformation languages* is the crucial factor, since transformation languages are for MDE as important as compilers are for high-level programming languages. Although numerous model transformation languages have been already proposed (for a survey, cf., [4]), currently no transformation language, not even the QVT (Query/View/Transformation) standard of the OMG [1], became accepted as the state-of-the-art model transformation language, i.e., an adoption in practice has not yet been achieved [7]. In particular, understandability and debuggability of model transformations are scarcely

supported by current approaches due to the following deficiencies. First, the artifacts involved in a model transformation, i.e., models, metamodels, as well as the actual transformation logic, are not represented in an *integrated view*. Instead, existing approaches only introduce formalisms for representing the transformation logic without considering the explicit representation of models and metamodels. Second, existing model transformation languages exhibit an inherent *impedance mismatch* between the specification and the execution of model transformations in terms of a one-to-many derivation of concepts. This is above all due to the fact that they do not support an explicit runtime model for the execution of model transformations which may be used to observe the runtime behavior of certain transformations [7], but rather execute their transformations on a low level of abstraction, e.g. based on a stack machine.

We therefore propose a novel formalism for developing model transformations called Transformation Nets [14], which tackles the aforementioned limitations of existing approaches. This formalism is based on Colored Petri Nets [8] and follows a process-oriented view towards model transformations. The main characteristic of the Transformation Net formalism is its ability to combine all the artifacts involved, i.e., metamodels, models, as well as the actual transformation logic, into one single representation. The possibility to gain an explicit, integrated representation of the semantics of a model transformation makes the formalism especially suited for gaining an understanding of the intricacies of a specific model transformation. This goes as far as to running the model transformation itself, as the Transformation Net constitutes a dedicated runtime model, thus serving as an execution engine for the transformation. This insight into transformation execution particularly favors the debugging and understanding of model transformations. Furthermore, Transformation Nets allow to build reusable modules that bridge certain kinds of structural heterogeneities, which are well-known in the area of database systems, as we have already shown in [10].

The remainder of this paper is structured as follows. Section 2 introduces the Transformation Net formalism. The subsequent Sections 3 and 4 present how the Transformation Net formalism is meant to be employed for concrete transformation problems by applying the formalism to a concrete example. Section 5 critically reflects the formalism by reporting on lessons learned from two case studies which have been conducted. Related work is discussed in Section 6. Finally, Section 7 gives a conclusion as well as an outlook on future work.

2 Transformation Nets at a Glance

This section introduces the Transformation Net formalism, whereby the conceptual architecture is shown in Figure 1. In this figure, we see a source metamodel on the left hand-side and a target metamodel on the right-hand side. In between, the transformation logic resides describing the correspondences between the metamodel elements. Furthermore, we see an input model conforming to the source metamodel, as well as an output model conforming to the target metamodel that represents the output of the transformation. The middle of Figure 1

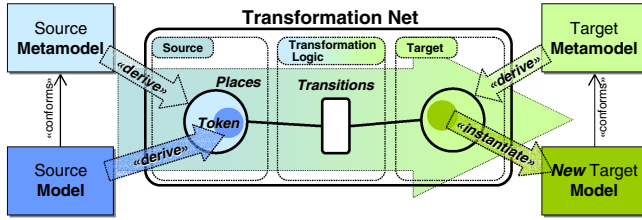


Fig. 1. Conceptual Architecture of Transformation Nets

shows the Transformation Net which represents the static parts of the transformation (i.e. metamodels and models) as places and tokens, respectively and the dynamic parts (i.e. the transformation logic) as appropriate transitions.

Transformation Net Metamodel. The abstract syntax of the Transformation Net language is formalized by means of a metamodel (cf. Figure 2) conforming to the Ecore meta-metamodel, the Eclipse realization of OMG’s MOF standard. This Transformation Net metamodel is based on Colored Petri Net concepts [8], but represents a specialized version thereof which aims at fulfilling the special requirements occurring in the domain of model transformations. In particular, in order to be able to encode metamodels and models, we introduce two kinds of places and two kinds of tokens (cf. Section 3). The second major adaption concerns the transitions. Since transitions are used to realize the actual transformation logic, we borrow a well established specification technique from graph transformation formalisms [6], which describe their transformation logic as a set of graphically encoded productions rules (cf. Section 4).

The whole Transformation Net metamodel is divided into four subpackages as can be seen in Figure 2. Thereby the package **Containers** comprise the modularization concepts. The package **StaticElements** is used to represent the static parts of a model transformation, i.e., metamodels and models. The dynamic elements, i.e., the actual transformation logic, are represented by concepts of the package **DynamicElements**. The package **Connectors** finally is responsible for connecting the static parts with the dynamic parts.

3 The Static Part of Transformation Nets

When employing Transformation Nets, in a first step, the static parts of a model transformation, i.e., the metamodels and models, need to be represented in our formalism (cf. package **StaticElements** in Figure 2). This incurs transitioning from the graph-based paradigm underlying MDE into the set-based paradigm underlying Petri Nets. The design rational behind this transition is the following: We rely on the core concepts of an object-oriented meta-metamodel, i.e., the graph which represents the metamodel consists of classes, attributes, and references, and the graph which represents a conforming model consists of objects,

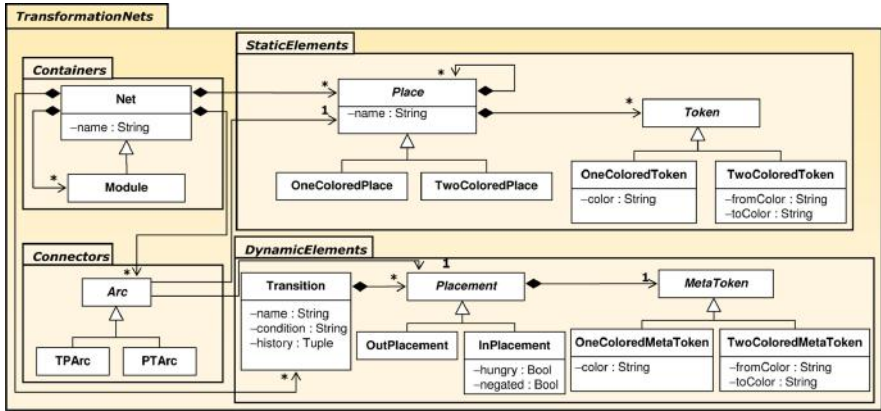


Fig. 2. The Transformation Net metamodel

data values and links. Therefore we distinguish between one-colored places containing one-colored tokens for representing the nodes of graphs, i.e., the objects, and two-colored places containing two-colored tokens. These two-colored tokens are needed for representing on the one hand links between the objects, i.e., one color represents the source object and the other the target object, and on the other hand attribute values, i.e., one color represents the containing object and the other the actual value.

Running Example. For describing the Transformation Net formalism in detail, we make use of a running example. The example is based on the Class2Relational case study¹ which became the de-facto standard example for model transformations, transforming an object-oriented model into a corresponding relational model. Due to reasons of brevity, only the most challenging part of this case study is described in this paper, namely how to represent inheritance hierarchies of classes within relational schemas.

Figure 3 shows UML diagrams of a simplified object-oriented metamodel as the source, and a metamodel for relational schemas as the target, together with a conforming source model and a to be generated target model. Thereby a ‘one-table-per-hierarchy’ approach is followed. Arguably, in terms of O/R mapping strategies, this may not be the most sophisticated approach. However, it makes the model transformation much more intriguing, thanks to the transitive closure that has to be computed over the class hierarchy. The middle layer of Figure 3 shows how the metamodel elements and model elements are represented as places and tokens, respectively, which is discussed in the following subsections.

3.1 Representing Metamodel Elements as Places

Classes Represented as One-Colored Places. Both, abstract and concrete classes are represented as `OneColoredPlaces`. Subclass relationships are

¹ <http://sosym.dcs.kcl.ac.uk/events/mtip05>

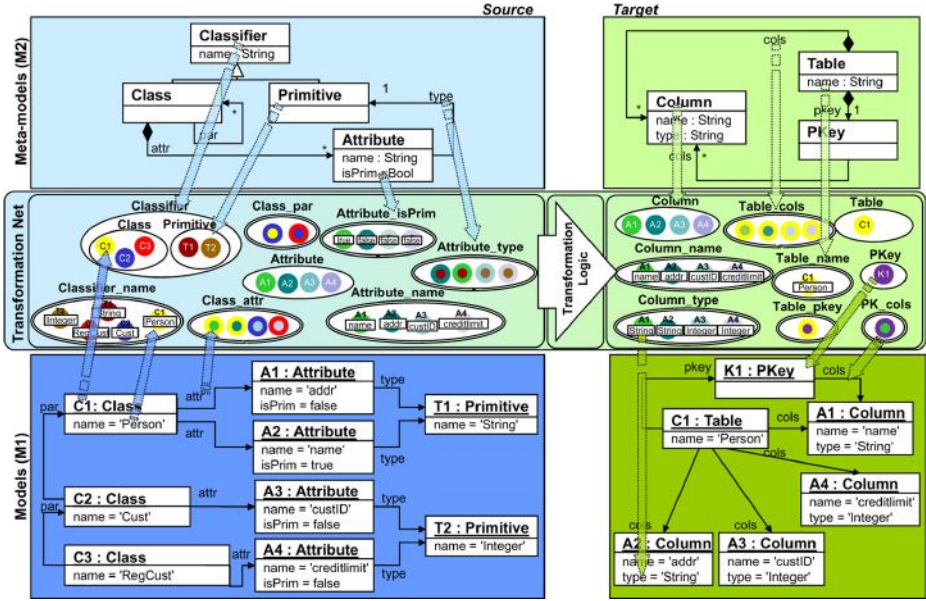


Fig. 3. The Class2Relational Transformation Problem

represented by the notion of places being contained within places. The notation used to visually represent one-colored places is a circle or an oval, which is traditionally used to depict places in Petri Nets. Concerning the example, depicted in Figure 3, one can see that each class of the metamodels – Classifier, Primitive, Class and Attribute of the source metamodel as well as Table, Column and PKey of the target metamodel – got represented through a respective one-colored place. Since Class and Primitive are subclasses of Classifier, these places became nested into the super-place.

References and Attributes Represented as Two-Colored Places. References and attributes are represented as TwoColoredPlaces. Notationally, two-colored places are represented like one-colored places. However, for easier distinction, indicating that these places contain two-colored tokens, the borders of two-colored places are doubly-lined. Considering the running example, one can see that for each reference like e.g. Attribute.type and for each attribute like e.g. Attribute.name a corresponding two-colored place – Attribute.type and Attribute.name, respectively – has been created.

3.2 Representing Model Elements as Tokens

Objects Represented as One-Colored Tokens. For every object, that occurs in a source model, a OneColoredToken is produced, which is put into the place that corresponds to the respective class in the source metamodel. The “color” is in fact expressed by means of a unique value that is derived from the

identifying attribute of the original model object. Hence, all one-colored tokens are “colored” with a unique literal. With regard to the running example, one can see that each instance of a class got represented through a respective one-colored token. Therefore, e.g. the one-colored place **Class** contains three one-colored tokens with distinct colors each one representing one of the three class instances **Person**, **Cust** and **RegCust**.

Links and Values Represented as Two-Colored Tokens. For every link as an instance of a reference, as well as for every value as an instance of an attribute, a **TwoColoredToken** is produced. The **fromColor** attribute of such a token (cf. Figure 2) refers to the color of the token that corresponds to the owning object. The **toColor** is given by the color of the token that corresponds to the linked target object or the primitive data value. Notationally, a two-colored token is represented as a ring (denoting the “from”-color) surrounding an inner circle (denoting the “to”-color). Concerning the example, one can see that for each link as well as for each value a two-colored token got generated. Therefore, e.g. the two-colored place **Class_{par}** contains two tokens, in which one of these represents the inheritance relationship between the class **Cust** and the class **Person** and the other one represents the inheritance relationship between the class **RegCust** and the class **Cust**.

4 The Dynamic Part of Transformation Nets

After the previous section dealt with describing how models and metamodels are represented as the static parts of a Transformation Net, this section introduces the dynamic parts of a Transformation Net. The actual transformation logic is embodied through a system of Petri Net transitions and additional places which reside in-between those places representing the original input and output metamodels as is shown for the **Class2Relational** example in Figure 4. In this way, tokens are streamed from the source places through the Transformation Net and finally end up in target places. Hence, when a Transformation Net has been generated in its initial state, a specialized Petri Net engine can then execute the process and stream tokens from source to target places. The resulting tokens in the places that were derived from elements of the target metamodel are then used to instantiate an output model that conforms to the target metamodel.

Matching and Producing Model Elements by Firing Transitions. An execution of a model transformation has two major phases. The first phase comprises the matching of certain elements of the source model from which information is derived that is used in the second phase for producing the elements of the output model. This matching and producing of model elements is supported within Transformation Nets by firing transitions. In Colored Petri Nets, the firing of a transition is based on a condition that involves the values of tokens in input places. Analogously, transitions in a Transformation Net are enabled if a certain configuration of matching tokens is available. This configuration is expressed with the remaining elements of the previously shown Transformation Net

metamodel (cf. subpackage *DynamicElements* in Figure 2). Thereby, transitions are represented through the **Transition** class. To specify their firing behavior, a mechanism well known from graph transformation systems [6] is used. Thereby, two patterns of input and output placeholders for tokens are defined, which represent a pre- and a post-condition by matching a certain configuration of tokens from the input places, and producing a certain configuration of tokens in the output places. The matching of tokens is the activity of finding a configuration of tokens from input places which satisfies the transition's pre-condition. Once such a configuration is found, the transition is enabled and ready to fire, with the colors of the input tokens to be bound to the input pattern. The production of output tokens once a transition fires, is typically dependent on the matched input tokens. For instance, when a transition is simply streaming a certain token, it would produce as an output the exact same token that was matched as an input token, thereby for example (cf. (c) in Figure 4) transforming an attribute of a top class into a column of the corresponding table.

Specification of Transition's Firing Rules. In general, transformation rules are more complex than just transforming one element of the source model into exactly one element of the target model. Instead, to resolve structural heterogeneities only specific elements fulfilling certain conditions have to be selected, computations for deriving new values and links have to be done, and completely new elements have to be generated in the target model which do not exist in the source model. Considering our running example, such a complex transformation rule is e.g. that only top classes should be transformed into tables. For describing complex transition firing rules, we have chosen the following specification mechanism (cf. Figure 2). A transition can have a number of **Placement** objects. Such a placement is merely a proxy for a certain input or output place which is connected to the placement by an **Arc** object. The incoming and outgoing arcs of a transition are represented by the classes **PTArc** and **TPArc**, which connect to its owned **InPlacement** and **OutPlacement** objects. Every placement can then contain a **MetaToken** object, represented in the metamodel through the class **MetaToken** and its specializations **OneColoredMetaToken** and **TwoColoredMetaToken**. Hence, a meta token can either stand for a one-colored or a two-colored token and can be used in two different ways:

- **Query Tokens:** Query tokens are meta tokens which are assigned to input placements. Query tokens can either stand for one-colored or two-colored token configurations, whose colors represent variables that during matching are bound to the color of an actual input token. Note that the colors of query tokens are not the required colors for input tokens, instead they describe color combination patterns that have to be fulfilled by input tokens. Normally, query tokens match for existence of input tokens but with the concept of negated input placements it is also possible to check for the non-existence of certain tokens. For example, this is required in our running example to find top classes, because a top class is a class which does not have an outgoing **par** link to another class.

- **Production Tokens:** Output placements contain so-called production tokens which are equally represented through the class `MetaToken` and its subclasses. For every production token in an output placement, a token is produced in the place that is connected to the output placement via an outgoing arc. The color of the produced token is defined by colors that are bound to the colors of the input query tokens. However, it is also possible to produce a token of a not yet existing color, for instance if the color of the output query token does not fit to any of the input query tokens. With this mechanism, new elements can be created in the target model which do not exist in the source model. Considering our running example, this mechanism is needed in order to produce primary keys in the target model which are not explicitly represented in the source model.

Please note that the default firing behavior of Transformation Nets is different to standard Petri Nets in the sense that transitions in standard Petri Nets always consume the tokens from the input places and produce new tokens in the output places. This behavior can also be achieved in Transformation Nets by setting the value of the attribute `hungry` of the corresponding `Inplacement` to “true”. It has to be emphasized, however, that this is not the default setting due to the fact that it is often the case that more than one transition has a certain place as an input place and therefore if all the connected transitions would consume the tokens, erroneous race conditions would appear. Therefore, by default, every transition is just reading the tokens of the connected input places and does not delete them. In order to prevent a transition to fire more than once for a certain token configuration, the already processed configurations are stored in a so-called switching history (cf. attribute `history` in Figure 2). In our running example, all transitions are marked as being not hungry.

Transformation Logic for the Class2Relational Example. To exemplify the use of transitions for defining transformation logic, Figure 4 depicts the transitions necessary in our running example. Thereby the transformation net is shown in it’s final state after streaming all the tokens through the net. As mentioned in Section 3, a ‘one-table-per-hierarchy’ approach is pursued resulting in the need for computing the transitive closure, i.e. making explicit all inheritance relationships, of the class hierarchy. Module (a) contains two transitions and a place which are jointly responsible for fulfilling this task. Thereby the left transition just copies the inheritance links defined in the model by matching a two-colored token from the input place `Class_par` and streaming this token to the connected output place. This output place accumulates all inheritance links including the transitive ones that are computed by the following right transition. This transition takes two two-colored tokens, each one representing a parent-link of the inheritance hierarchy and if the inner color of the one input-token matches the outer color of the other input token, i.e., there is another class that inherits from the subclass, a link is established from this indirect subclass to the super-class and put into the corresponding place. In this way, all possible inheritance relationships can be derived. The ones that have no further parent are extracted

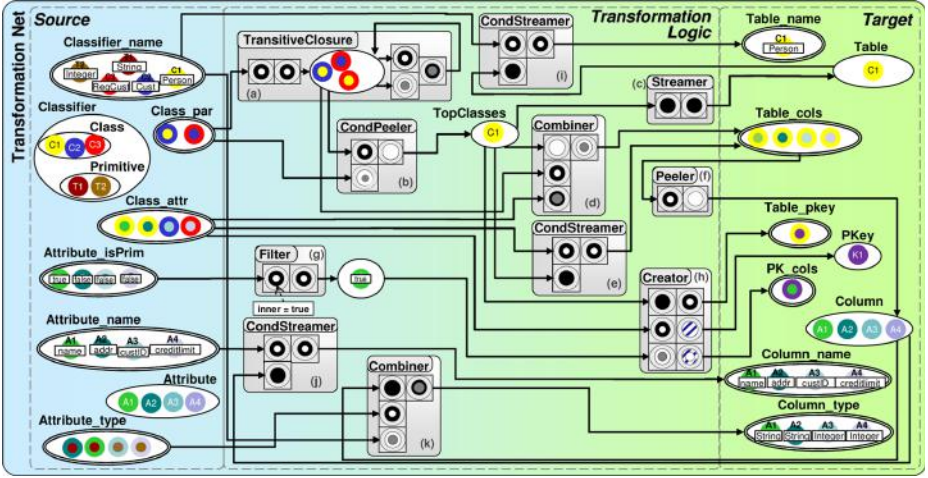


Fig. 4. Class2Relational example realized with Transformation Nets

and matched by the transition of module (b). Note that for this transition we have to use a negated input placement represented by a dashed circle. If such a matching token configuration is found, the transition takes the inner color of this link and streams it to the **TopClasses** place, since such a token represents a top-level-class of the inheritance hierarchy. These top-level-classes are of special interest in this transformation as the number of connected transitions to this place reveals. Module (c), for instance, is responsible for creating tables for each found top-level class and therefore just streams tokens of this place to the **Table** place. Modules (d) and (e) are responsible for computing the columns of the generated tables and therefore also rely on the top-level classes. In order to accomplish this task, module (e) streams all those **Class_attr** tokens to the **Table_cols** place that are owned by a top-level class. Additionally since a ‘one-table-per-hierarchy’ approach is followed, also those attributes need to become columns which are contained in some subclass and for this task module (d) is responsible. Thereby all those **Class_attr** tokens are streamed to the **Table_cols** place which are in a direct or indirect inheritance relationship to a top-level-class according to the transitive closure. From these two-colored tokens, module (f) generates tokens for the **Column** place by peeling the inner color out of the two-colored tokens and generating one-colored output tokens. For generating primary keys from identifier attributes, i.e., attributes where **isPrim = true**, module (g) and (h) are employed. While the first one is a special kind of conditional streamer for filtering the identifier attributes by evaluating the condition (**inner = true**) and assigning the result token to the transition instead of an additional query token, the second one is responsible for generating for each identifier attribute a new one-colored token representing a primary key and linking this newly created token with tables and columns accordingly. Finally, module (i), (j), and (k)

are used to stream the attribute values for the previously generated tables and columns into the places `Table_name`, `Column_name`, and `Column_type`.

5 Lessons Learned

This section presents lessons learned from the Class2Relational case study. Additionally to this horizontal (i.e., model to model) transformation scenario, a vertical (i.e., model to code) transformation scenario, has been conducted in order to clarify the value of our approach for diverse application areas. The vertical scenario is the BPMN2BPEL example taken from a graph transformation tool contest². The case studies have been realized with the help of our prototype for modeling, executing and debugging transformation nets. Further details of the case study realizations and tool support may be found at our project page³.

Composition and weak typing favors reusability. First of all, it has been shown that several kinds of transitions occur many times with minor deviations only. Such transitions can be generalized to transformation patterns and subsequently realized by modules. Since the inplacements as well as the outplacements are just typed to one-colored tokens and two-colored tokens, respectively and not to certain metaclasses, these modules can be reused in different scenarios. This kind of reuse is not restricted to single transitions only, since through the composition of transitions by sequencing as well as nesting the resulting modules, modules, realizing complex transformation logic, can be built. The Class2Relational case study was realized by the usage of just eight different modules, whereby the `CondStreamer` module was applied three times (cf. Figure 4), thus justifying the potential for reuse.

Visual syntax and live programming fosters debugging. Transformation nets represent a visual formalism for defining model transformations which is especially favorable for debugging purposes. This is not least since the flow of model elements undergoing certain transformations can be directly followed by observing the flow of tokens whereby undesired results can be detected easily. Another characteristic of transformation nets, that fosters debuggability, is live programming, i.e., some piece of transformation logic can be executed and thus tested immediately after definition without any further compilation step. Therefore, testing can be done independently of other parts of the Transformation Net by just setting up a suitable token configuration in the input places.

Implicit control flow eases evolution. The control flow in a transformation net is given through data dependencies between various transitions. As a consequence, when changing a transformation, one needs to maintain a single artifact only instead of requiring additional efforts to keep control flow and transformation logic (in the form of rules) synchronized. For instance, when a certain rule would need to be changed to match for additional model objects, one has to explicitly take care to call this rule at a time when the objects to be matched already exist.

² <http://www.fots.ua.ac.be/events/grabats2008>

³ <http://big.tuwien.ac.at/projects/tropic>

Fine-grained model decomposition facilitates resolution of heterogeneities. The chosen representation of models by Petri nets lets references as well as attributes become first-class citizens, resulting in a fine-grained decomposition of models. The resulting representation in combination with weak typing turned out to be especially favorable for the resolution of structural heterogeneities. This is since on the one hand there are no restrictions, like a class must be instantiated before an owned attribute and since on the other hand e.g. an attribute in the source model can easily become a class in the target model by just moving the token to the respective place.

Transitions by color-patterns ease development but lower readability. Currently the precondition as well as the postcondition of a transition are just encoded by one-colored as well as two-colored tokens. On the one hand, this mechanism eases development since e.g. for changing the direction of a link it suffices just to swap the respective colors of the query token and the production token. On the other hand, the case study has shown that the larger the transformation net grows the less readable this kind of encoding gets. Therefore, it has been proven useful to assign each input as well as each output placement a human-readable label, that describes the kind of input and output, respectively.

6 Related Work

One of the main objectives of transformation nets is to enhance the debuggability and understandability of model transformations by using a Petri net based formalism. Therefore, we consider two orthogonal threads of related work. First, we discuss how debugging and understandability in terms of a visual representation as well as the possibility for graphical simulation are supported by current model transformation approaches, and second, we elaborate on the general usage of Petri Nets in model transformation approaches.

Debugging Support and Understandability. Debugging support only at the execution level requires traceability to the corresponding higher-level mapping specifications in order to be aware of the effects a certain mapping induces on the state of the execution. For example, in the Fujaba environment⁴, a plugin called MoTE [16] compiles TGG rules [11] into Fujaba story diagrams that are implemented in Java, which obstructs a direct debugging on the level of TGG rules. Additional to that, Fujaba supports visualization of how the graph evolves during transformation, and allows interactive application of transformation rules. Furthermore, approaches like VIATRA [2] producing debug reports that trace an execution, only, are likewise considered inadequate for debugging since a minimum requirement for the debugging should be the ability to debug at least whole transformation rules, by which we refer to as the stepwise execution and inspection of the execution state. The debugging of ATL [9] is based on the step-wise execution of a stack-machine that interprets ATL byte-code, which

⁴ <http://www.fujaba.de>

also allows observing the execution of whole transformation rules. SmartQVT⁵ [1], TefKat [12] and KerMeta [13] allow for similar debugging functionality.

What sets transformation nets apart from these approaches is that all debugging activities are carried out on a single integrated formalism, without needing to deal with several different views. Furthermore, this approach is unique in allowing interactive execution not only by choosing “rules” or by manipulating the state directly, but also by allowing to modify the structure of the net itself. This ability for “live”-programming enables an additional benefit for debugging and development: one can correct errors (e.g., “stucked” tokens) in the net right away without needing to recompile and restart the debug cycle.

Concerning the understandability of model transformations in terms of a visual representation and a possibility for a graphical simulation, only graph transformation approaches like, e.g., Fujaba allow for a similar functionality. However, these approaches neither provide an integrated view on all transformation artifacts nor do they provide an integrated view on the whole transformation process in terms of the past state, i.e., which rules fired already, the current state, and the prospective future state, i.e., which rules are now enabled to fire. Therefore, these approaches only provide snapshots of the current transformation state.

Petri Nets and Model Transformations. The relatedness of Petri nets and graph rewriting systems has also induced some impact in the field of model transformation. Especially in the area of graph transformations some work has been conducted that uses Petri nets to check formal properties of graph production rules. Thereby, the approach proposed in [15] translates individual graph rules into a place/transition net and checks for its termination. Another approach is described in [5], which applies a transition system for modeling the dynamic behavior of a metamodel.

Compared to these two approaches, our intention to use Petri nets is entirely different. While these two approaches are using Petri nets as a back-end for automatically analyzing properties of transformations, we are using Petri nets as a front-end for fostering debuggability and understandability. In particular, we are focussing on how to represent model transformations as Petri Nets in an intuitive manner. This also covers the compact representation of Petri Nets to eliminate the scalability problem of low-level Petri nets. Finally, we introduce a specific syntax for Petri Nets used for model transformations and integrate several high-level constructs, e.g., inhibitor arcs and pages, into our language.

7 Conclusion and Future Work

In this paper, we have presented the Transformation Net formalism which is meant to be a runtime model for the representation of model transformations. First investigations have shown that the formalism is promising to solve a wide spectrum of transformation problems like horizontal transformation scenarios and vertical transformation scenarios, respectively. Especially the debugging of model transformations is fostered since Transformation Nets provide an

⁵ <http://smartqvt.elibel.tm.fr>

integrated view on all transformation artifacts involved as well as a dedicated runtime model. For future work we strive to investigate formal properties like reachability, liveness or boundedness of Petri Nets and their potential applicability as well as usefulness for model transformations. Furthermore we aim at translating existing model transformation languages into transformation nets like the QVT-Relations standard. By doing so, we gain (1) operational semantics for the QVT-Relations standard and (2) a visual debugging possibility.

References

1. Object Management Group (OMG). Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification, Final Adopted Specification (2007)
2. Balogh, A., Varró, D.: Advanced model transformation language constructs in the VIATRA2 framework. In: 21st ACM Symposium on Applied Computing (2006)
3. Bézin, J.: On the Unification Power of Models. *Journal on Software and Systems Modeling* 4(2) (2005)
4. Czarnecki, K., Helsen, S.: Feature-based survey of model transformation approaches. *IBM Systems Journal* 45(3) (2006)
5. de Lara, J., Vangheluwe, H.: Translating Model Simulators to Analysis Models. In: Fiadeiro, J.L., Inverardi, P. (eds.) *FASE 2008*. LNCS, vol. 4961, pp. 77–92. Springer, Heidelberg (2008)
6. Ehrig, H., Engels, G., Kreowski, H.-J., Rozenberg, G. (eds.): *Handbook of graph grammars and computing by graph transformation. Applications, languages, and tools*, vol. 2. World Scientific Publishing Co., Singapore (1999)
7. France, R., Rumpe, B.: Model-driven Development of Complex Software: A Research Roadmap. In: 29th Int. Conf. on Software Engineering (2007)
8. Jensen, K.: *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use*. Monographs in Theoretical Computer Science. Springer, Heidelberg (1992)
9. Jouault, F., Kurtev, I.: Transforming Models with ATL. In: Bruehl, J.-M. (ed.) *MoDELS 2005*. LNCS, vol. 3844, pp. 128–138. Springer, Heidelberg (2006)
10. Kappel, G., Kargl, H., Reiter, T., Retschitzegger, W., Schwinger, W., Strommer, M., Wimmer, M.: A framework for building mapping operators resolving structural heterogeneities. In: Kaschek, R., et al. (eds.) *UNISCON 2008*. LNBIP 5, pp. 158–174 (2008)
11. Koenigs, A.: Model Transformation with Triple Graph Grammars. In: *Model Transformations in Practice Workshop of MODELS 2005* (2005)
12. Lawley, M., Steel, J.: Practical Declarative Model Transformation with Tefkat. In: *Model Transformations in Practice Workshop of MODELS 2005* (2005)
13. Muller, P., Fleurey, F., Jezequel, J.: Weaving Executability into Object-Oriented Meta-languages. In: 8th Int. Conf. on Model Driven Engineering Languages and Systems (2005)
14. Reiter, T., Wimmer, M., Kargl, H.: Towards a runtime model based on colored Petri-nets for the execution of model transformations. In: *Proceedings of the 3rd Workshop on Models and Aspects @ ECOOP* (2007)
15. Varró, D., Varró-Gyapay, S., Ehrig, H., Prange, U., Taentzer, G.: Termination Analysis of Model Transformations by Petri Nets. In: Corradini, A., Ehrig, H., Montanari, U., Ribeiro, L., Rozenberg, G. (eds.) *ICGT 2006*. LNCS, vol. 4178, pp. 260–274. Springer, Heidelberg (2006)
16. Wagner, R.: Developing Model Transformations with Fujaba. In: 4th Int. Fujaba Days (2006)

Metamodeling Foundation for Software and Data Integration

Henning Agt, Gregor Bauhoff, Mario Carlsburg, Daniel Kumpe, Ralf Kutsche,
and Nikola Milanovic

Technische Universität Berlin

{hagt,gbauhoff,mcartsbg,dkumpe,rkutsche,nmilanov}@cs.tu-berlin.de

Abstract. We propose a model-based methodology for integration of heterogeneous distributed systems, based on the multi-level modeling abstractions, automated conflict analysis and connector code generation. The focus in this paper is on the metamodeling foundation necessary for this process, and consequently we introduce computation independent, platform specific, platform independent and semantic metamodels, which generate a set of domain specific languages used to describe software and data integration scenarios.

1 Introduction

Integration of heterogeneous distributed IT-systems is one of the major problems and cost-driving factors in the software industry today [18]. The problem is not new, and several solution possibilities exist. Schema matching approaches [19] try to detect dependencies and conflicts between data model elements at the model or instance levels. Extract-Transformation-Load (ETL) tools use schema matching methodology to enable easier integration of multiple data sources. Many languages exist that enable specification of transformations between two data models, such as Ensemble [9]. The usability of all mentioned approaches suffers greatly with increased heterogeneity of the underlying data sources.

The second major group of integration approaches is focused on the Service Oriented Architecture (SOA). In SOA, all system interfaces are wrapped as service endpoints and accessible to the Enterprise Service Bus (ESB) engine, which orchestrates data and functional logic. However, it is expected that all service endpoints are compatible and no data or behavior conflicts will occur. If they do, either the endpoint itself has to be modified (frequently impossible), or the XSLT or Java code snippet (BPELJ) has to be written, correcting the conflict at the message level. There are approaches which fill the niche between two main groups such as mapping editors (e.g., Altova MapForce) and extended UML Editors (e.g., E2E Bridge), with very low or no support for the compositional conflict analysis and the code (endpoint) generation.

For these reasons, as a part of the R&D program of the German government for regional business initiatives, the project BIZYCLE (www.bizycle.de) was started in early 2007 in order to investigate in large-scale the potential of model-based software and data integration methodologies, tool support and practical

applicability for different industrial domains¹. The BIZYCLE integration process [12], [13], [17] is based on multi-level modeling abstractions. The integration scenario is first modeled at the computation independent (CIM) level, where business aspects of an integration scenario are described. The model is then refined at the platform specific (PSM) level, where technical interfaces of the systems that should be integrated are described. The PSM interface descriptions are then abstracted to platform independent (PIM) level, where a conflict analysis process takes place. Based on the result of the conflict analysis, connector model and code are generated and deployed.

As the metamodels behind the integration process are our focus, this paper is structured as follows: first the CIM metamodel is described, followed by one selected PSM metamodel and PIM metamodel. After that, we propose a way to semantically annotate model elements using semantical metamodel. Finally, we show how, based on the proposed metamodels, conflict analysis and connector code generation may be performed.

2 Computation Independent Metamodel

The BIZYCLE integration process captures early stages of an integration scenario at the computation independent model (CIM) level. It describes scenario requirements with an abstract business process and data flow model, regardless of the technical details of the underlying systems. Furthermore, it forms a basis for the conflict analysis which identifies interface interoperability mismatches.

The integration scenario is described in terms of activities and transitions, as well as data exchange aspects and their hierarchical structure (see Figure 1). **BusinessComponent** represents a software or data artifact that is involved in the scenario. Business components do not necessarily represent exactly one physical system, e.g., functional access can be provided as a Web service and data access with an SQL interface. A special business component is the **BusinessConnector**, which handles interconnections between the components. The abstract meta-classes **ActivityNode** and **ActivityEdge** are used to express data and control flow of the integration scenario.

Business components perform different activities. **ControlNodes** are used to determine the beginning and the end of an integration scenario. **ExportInterface** or **ImportInterface** are used to describe the terms of data exchange or function calls. Activities of the type **InternalComponentAction** express actions that are not relevant to the concrete data flow, but are useful for understanding the context of the scenario. The metamodel also defines **BusinessFunction** that represents functionality of the integrated artifacts. It processes incoming data or performs a certain task required by other artifacts. Business functions can only be used inside of business components. **ConnectorFunction** is used to predefine functionality of the connector. It can represent control (e.g., **Timer**) or conflict resolution functions (e.g. **Transformer**, **Router** or **Aggregator**), as in [8].

¹ This work is partially supported by the Federal Ministry of Education and Research (BMBF) under grant number 03WKBB1B.

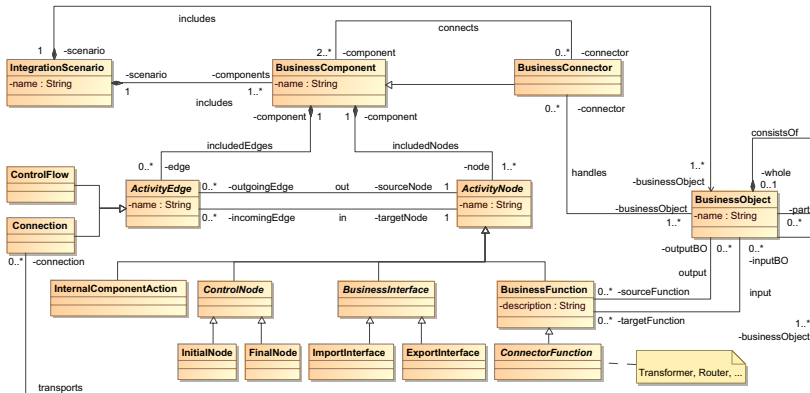


Fig. 1. CIMM – Basic metaclasses for artifact, process and data modeling (excerpt)

Two different kinds of edges can connect activities of the integration process. The **ControlFlow** is used to express the transition from one activity to another inside of a business component. The **Connection** models data and control flow between different business components or connectors and transports **BusinessObject** elements, which can be hierarchically structured independent of any data type of the underlying interfaces. Business objects are produced or consumed by **BusinessFunctions**. Constraints in the metamodel assure the correct combination of activity edges and nodes.

Four CIM views are defined that graphically show relevant aspects of the model: flow, object, connection and function view. The CIM flow view shows the sequence of the integration activities and transitions, similarly to the UML activity diagram. The object view consists of the business object and their structure only. The connection view hides the internal behavior of the components and displays business objects related to their connections. Finally, the function view shows business functions and their input and output business objects.

3 Platform Specific Metamodels

Platform specific metamodels (PSMM) describe the structure, behavior, communication and non-functional properties of the platform-specific system interfaces. The communication part describes information required to establish a connection with the system. The structure part provides information about platform-specific type system. The property part describes non-functional properties, such as performance, authorization, security or logging. It is also possible to annotate model elements using concepts from an existing ontology (see Section 5).

Ecore serves as a metamodel for the PSMMs [2]. Consequently the PSMMs are at the M2-level of the MOF hierarchy. The elements of the PSMMs are linguistic instances (see [11]) of the Ecore metamodel. The PSMMs define a domain specific language (DSL) for the description of software system interfaces at the

M1 level. To be more precise, the elements of PSMMs build the vocabulary for the interface descriptions. The system under study is provided on the M0-level. Every metamodel-layer describes only concepts for one layer beneath, therefore one has a linear metamodel hierarchy over four semantic abstraction levels, conforming to the MOF-hierarchy. An overview about the different metamodel layers is given in [6] and different metamodeling roles are defined in [11].

We currently provide metamodel support for the following platforms: ERP systems (SAP R/3 and AvERP), relational and XML databases, Web Services, XML files, J2EE components and .NET applications. As an illustration, in the following subsection we describe the SAP R/3 metamodel in more detail. All other PSMMs have a similar package composition to provide adequate DSLs for the remaining platforms.

3.1 SAP R/3 Platform Specific Metamodel

SAP R/3 is one of the most frequently used enterprise resource planning (ERP) systems. The platform specific metamodel of SAP R/3 consists mainly of classes referring to the remote accessible methods, so-called BAPIs (Business Application Programming Interface) and IDOCs (Intermediate Document). BAPIs are remote accessible functions and IDOCs are structured files. The schema of an IDOC file is defined within an IDOC Type. The IDOC type consists of a hierarchic tree with segments and fields inside the segments. The entry point to the hierarchic tree-based model is the element SAP R3, which represents a concrete SAP R3 system installation. The SAP R3 consist of at least one SAP R3 Interface. This element consist of the elements Access, SAP Business Component and IDOC Type. The whole parameter and structure part can be built automatically through extraction of relevant information from the SAP Business Object Repository (BOR). The core metamodel (excerpt) is shown in Figure 2.

The elements Communication Channel and Access belong to the package communication. This package collects information about the physical access, including user name, password, host name, language, system number, sap client and the communication channel. A communication channel can be synchronous, asynchronous, transactional or queued RFC (remote function control), simple file transfer, or e-mail. The concrete communication channel depends on the type of information exchange. IDOCs will be exchanged over file transfer, e-mail or transactional RFC, and BAPIs will be accessed using synchronous RFC.

Parameters can have the structured, table and field types. Structured and table types can consist of field types, which are atomic. Field types are JCO (Java Connector) types, more precisely they are wrapper classes. JCO offers a library for Java which supports the access to a SAP R3 system. JCO converts the SAP R3 types to Java types and backwards.

The abstract element Method is linked to the package Behavior and the element SAP R3 is linked to the package Property. For more information about behavior and properties, see section 4 about platform independent metamodel.

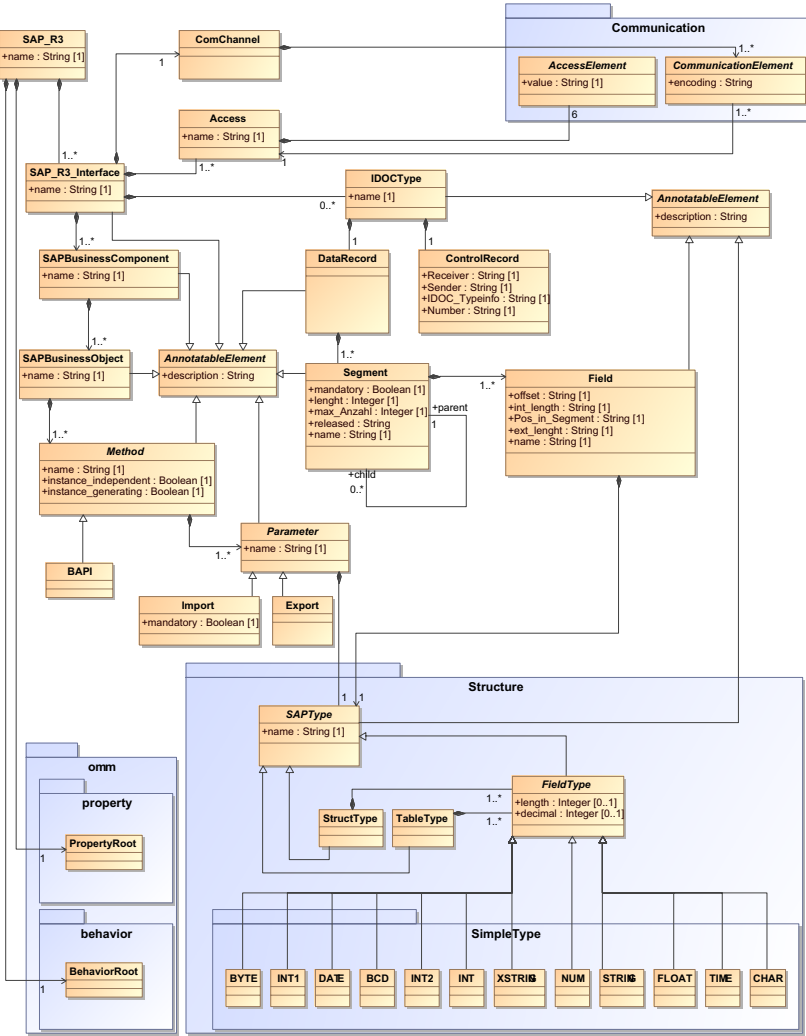


Fig. 2. Excerpt from Core SAP R/3 metamodel

Parameters and segments have references to the domain object elements of the semantic metamodel (see Section 5) and can therefore be linked with domain object within an ontology. Likewise, all methods can be linked to domain functions.

4 Platform Independent Metamodel

The purpose of the platform independent metamodel (PIMM) is to facilitate system interoperability by abstracting all platform specific heterogeneous interface details. The abstraction process is realized by a PSM-to-PIM transformation

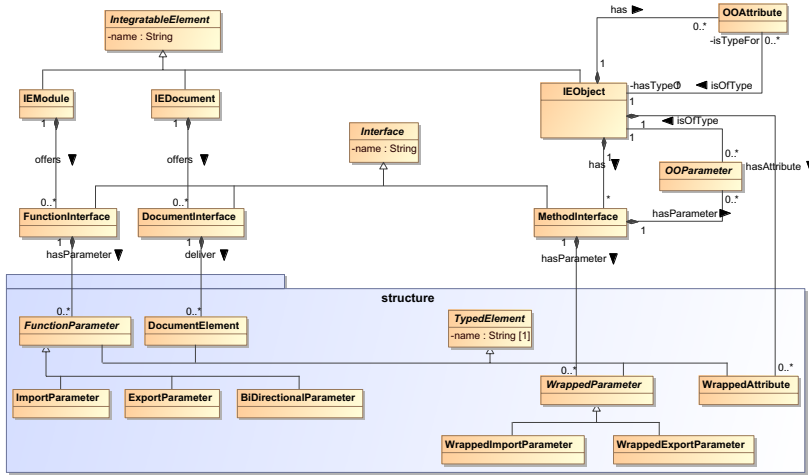


Fig. 3. PIMM - basic metaclasses for interfaces (excerpt)

using ATL [1]. For every PSMM there exists a set of transformation rules which translates the PSM into the common abstraction layer, the PIM. As stated in [14] the PIMM facilitates integration of heterogeneous interfaces. At the PIM level it is possible to represent different interface details on a common basis.

At the PIM level an **Interface** represents a single system gateway which is able to handle data as input and/or output in one single step. Hence PIMM-Interfaces represent an abstraction for all platform specific operations, methods, functions, files etc. PIMM distinguishes between three interface types: **FunctionInterface**, **MethodInterface** and **DocumentInterface**. The first represents a non-object-oriented interface, the second an object-oriented interface and the last a data structure-based interface. All interfaces have different corresponding metaclasses: DocumentInterfaces have only one root **Type** and no parameters, FunctionInterfaces can have different **FunctionParameters**, each with a corresponding type and MethodInterfaces can have object-based parameters with object-identity represented by **IEObject**. Due to this fact, our design decision was not to generalize everything into one single interface type. All interface types are still platform independent from a technical point of view.

Every interface has its own parent container, the system which exposes it, modeled by **IntegratableElement**. Each interface contains associations to different parts of the PIMM which will be described in the following subsections (see basic metaclasses in Figure 3).

4.1 Structure and Communication

The structure part (Figure 4) includes the common type system with the abstract super metaclass **Type** and different sub-metaclasses to express **SimpleTypes** (String, Number, Boolean) as well as **ComplexTypes**. The abstract **TypedElement**

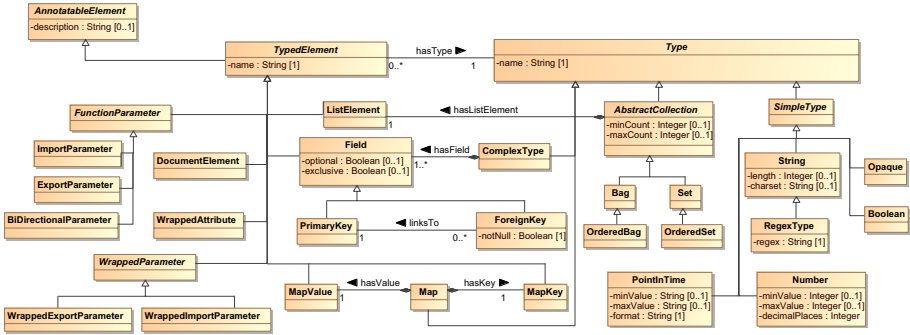


Fig. 4. PIMM - structure part

represents an element with a specific meaning (semantics) and represents a value container at runtime, e.g. **ImportParameter**. A **TypedElement** has exactly one association to a type but vice versa a type can belong to arbitrarily **TypedElements**, which means a **Type** can be 'reused'. For each interface type, different sub-metaclasses of **TypedElement** are provided: **WrappedParameter**, **FunctionParameter** and **DocumentElement**.

An interface may have its own interaction rules. These can be described by communication patterns such as **MessageExchangePatterns**, or different **Call**- and **Event**-types. Knowing platform independent communication details is essential to be able to interact with every interface in the proper way. During communication conflict analysis, incompatible call mechanisms are detected and fixed (as far as possible automatically).

4.2 Property and Behavior

Non-functional properties are used to characterize interface capability (provided) and expectations (required) properties. The root node **PropertyRoot** is responsible for including every **PropertyContainer** which is derived to **RequirementPC** and **CapabilityPC** element. Every **PropertyElement** is a child of exactly one of these two sub-metaclasses. The metaclass **QualityOfService** is the upper metaclass of every QoS property and has an association to a **Metric**. All metrics have the **AbstractMetric** as their upper metaclass. Based on this metamodel construction every QoS can be combined with every metric. During property conflict analysis, **CapabilityPCs** and **RequirementPCs** are related to each other in order to extract incompatible process flow graphs.

Behavior of a system is described using OCL and process algebra metaclasses. The OCL can be used to define parameter and function constraints: conditions between parameter values, pre-conditions, post-conditions and invariants using abstract states. Process algebra interface call order describes the allowed call behavior from a client side view. The connector (mediator) acts as a client and calls different interfaces (e.g., methods) sequentially or concurrently.

5 Semantic and Annotation Metamodel

One of the challenges in software and data integration projects is the (semi)-automatic detection of mismatches in interface semantics. A prerequisite for the semantic analysis is the semantic description of the integration scenario at CIM and PSM/PIM abstraction levels. Our current solution follows the approach of a shared domain ontology [20] to subsume the semantic descriptions that can be used in several integration projects.

The semantic metamodel SMM (Figure 5) allows to create a graph of triples in terms of subject, predicate, and object relations, similar to RDF. Subjects and objects are modeled with the **SemanticConcept** metaclass that can be either a **DomainObject** (knowledge representation of data) or a **DomainFunction** (knowledge representation of functionality). Semantic concepts are linked with **Predicates**. The metamodel includes predefined predicates which we identified as relevant for knowledge relations in integration scenarios: generalization (**IsA**), data processing (**Input**, **Output**), containment (**Has**) and sets (**ListOf**). The **CustomPredicate** allows the creation of user specific relations.

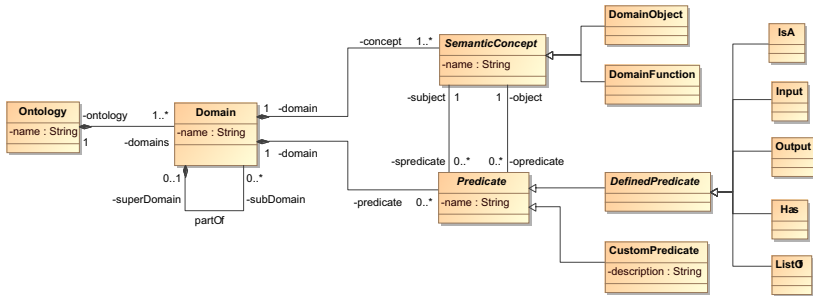


Fig. 5. Semantic metamodel

The association of heterogeneous artifacts such as documents, service interfaces, business processes, web resources and models with semantic concepts is called semantic annotation [10, 4]. The goal of the semantic annotation is a distinct semantic characterization of abstract data definitions, underlying interfaces and interface element structures to enable a (semi)-automatic conflict analysis and resolution. We distinguish between data-oriented annotations and function-oriented annotations.

The annotation metamodel (AMM) offers both kinds of annotations. They are realized by a model weaving technique [5]. Figure 6 shows an excerpt of the AMM for J2EE systems to exemplary illustrate the annotation capabilities. The AMM offers the **DomainFunctionAnnotation** metaclass to link the functional parts of the platform specific metamodel with the domain function of the semantic metamodel. On the right side of the picture, all data related elements of the J2EE PSM are linked to the semantic metamodel using the

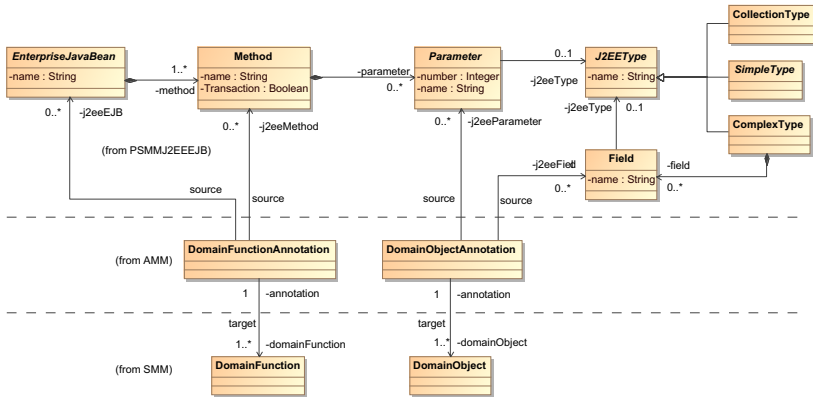


Fig. 6. Excerpt of the annotation metamodel - J2EE annotations

DomainObjectAnnotation. In addition, the AMM contains all further associations to other annotateable metamodel elements (AMM is implemented as Ecore model with EReferences to all the other Ecore models). One of the main advantages of semantic annotation through all levels of abstraction is the traceability of information. Apart from the annotation metaclasses, the AMM also offers means to combine annotations with logical operators. With the operators and varied use of single or multiple references it is possible to build containment, choice and multiple representation expressions.

6 Conflict Analysis

The metamodels forming DSL foundation at CIM, PSM and PIM levels are used for two main purposes: conflict analysis of interface mismatches and connector code generation. The conflict analysis algorithm examines interface descriptions at the PIM level in conjunction with the abstract process and data requirement definitions at the CIM level.

The *semantic* analysis uses the ontology and semantic annotations to check whether the abstract data flow requirements at the CIM level can be fulfilled by interface descriptions at the PSM/PIM level. Additionally, it determines dependencies of interface parameters and verifies their functional annotations according to the business function definitions in the CIM model. Using logical reasoning, mismatched requirements are resolved if possible. The results of the analysis are requirement mappings of the abstract business objects to exporting and importing interfaces elements at the PSM/PIM level. An in-depth description of the semantic annotation and conflict analysis can be found in [3].

The *behavior* analysis derives a first interface call order. Dependencies and process definitions at the CIM level are checked against the behavior constraints of the interfaces, such as pre- and post-conditions. Closely related is the *property*

analysis that examines interfaces' QoS properties and metrics, such as WCET or reliability [16]. The result of both analyses is a refined interface call order. The *communication* analysis then takes into account characteristics of interface interaction [15]. The results of this analysis, together with the information from the PSM level, are used to generate application endpoints that communicate with the system and offer a common access pattern to the connector.

Finally, the *structure* analysis overcomes the structural heterogeneity of the interfaces by performing range of values comparison, identifying required data type converters as well as creating a message processor lists for the e.g., merge, split or filter of data structures. The work of integration specialists is hence supported to a certain degree of automation, but due to the complexity of software systems a fully automated conflict resolution is difficult to accomplish in general case. Remaining conflicts or multiple conflict-free choices are resolved manually.

7 Connector Generation

Based on the models describing the integration scenario and results of the conflict analysis, the connector component model and code are generated. A connector is an automatically generated component, which is used to overcome all discovered conflicts and enable technical, semantical and business interoperation. It is based on the principles of message oriented middleware (MOM), and its metamodel is accordingly based on the message passing.

The connector generation starts with the **ChannelAdapter** which comprises **ApplicationEndpoint** and **MessageGateway**. **ApplicationEndpoint** implements the technical interoperability, and is able to call remote system interfaces. It passes export parameters to the **MessageGateway**, which serializes them into **Message**. In the other direction, **MessageGateway** deserializes a message and passes it to the **ApplicationEndpoint**. Messages are further transported by **Channels** which can be either 1-1 channels or publish/subscribe. **MessageProcessors** perform conflict resolution by executing aggregation, routing, transformation, enrichment etc. functions. Application endpoints can be generated using standard code generation methods or using model interpretation. Core connector logic (**Channels** and **MessageProcessors**) are interpreted based on the UML Action Semantics description.

The code generation approach is based on Java Emitter Template (JET). The first step is to read all import and export parameters and create Java classes out of them, where each parameter is wrapped within one class. In this step, PSMM types are transformed to the PIMM (Java) types. The second step is to create a Java method for each function modeled in the PSM. This method encapsulates access to the target system. Import parameters are sent to the application endpoint as a hash map, equivalently, export parameters are also received as a hash map. Alternative is to use model interpretation, where application endpoints are built with Java code at runtime, starting with the interpretation of the platform specific model using the model interpreter component. It reads the Ecore-file of an interface and initiates a connection. Afterwards the interpreter builds a

Java object, which implements the Interface *ICallableObject*. It consists of executable functions which conform to the PSM. The endpoint gateway component is responsible for receiving and sending of messages. It provides an interface with methods for registration of the business connector, in order to support the data transfer. Data converter supports translation of data into different formats, allowing a lossless and smooth data exchange within an application endpoint.

UML Action Semantics models are used to describe internal behavior of the connector core components. We extended basic UML Action Semantics meta-model to allow for the following action types: string, mathematical, logical, date and time and code actions (as extensions of computation actions), as well as read and write, type conversion, composite and collection actions [7]. Using this vocabulary, patterns such as Transformer, Normalizer or Aggregator are built, which are then interpreted as Java code using Java Message Service (JMS).

Using the connector metamodel and code generation techniques, it is possible to exchange the underlying runtime environment without any manual intervention, for example, instead of JMS gateways and Java transformation logic, Web service (WSDL) gateway, BPEL orchestration code and XSLT transformations for the SOAP messages can be generated.

8 Conclusion

The project research results have been prototypically realized as the BIZYCLE Model-Based Integration Framework (MBIF). It is based on the presented meta-models to support different abstraction levels that are part of the BIZYCLE integration process. Beside the modeling platform, which guides a developer through the integration process steps, MBIF consists of two other main components: BIZYCLE Repository (offers all needed persistence services, version and consistency management) and BIZYCLE Runtime Environment (where the generated business connectors are deployed and executable models interpreted).

Based on the practical experience and feedback from our industrial partners, several benefits can be already identified. An important aspect is a degree of automation, achieved through code generation, systematic conflict analysis process and automated technical model extraction. Reuse is supported at the model-level, as interface descriptions, transformation rules and semantic annotations can be shared between multiple projects via BIZYCLE Repository. The evolution is supported at the model level, and code generation methods enable smooth transitions. Metamodeling enables very fast tool prototyping. However, metamodels also improve understanding of the problem domain. One of the major advantages of the proposed solution are multiple abstraction levels, such as CIM, PSM, PIM and code, which enable business architects not to start at the data model and/or code level right away, as is usual in today's practice. The essential benefit offered by the multi-level modeling environment is based on the capability of performing (to a high degree) automated model transformations, abstracting and refining over the given level hierarchy.

References

1. ATL: Atlas Transformation Language User Manual (2006), http://www.eclipse.org/m2m/at1/doc/ATL_User_Manualv0.7.pdf
2. Eclipse modeling framework (2008), <http://www.eclipse.org/modeling/emf/>
3. Agt, H., Widiker, J., Bauhoff, G., Milanovic, N., Kutsche, R.: Model-based Semantic Conflict Analysis for Software- and Data-integration Scenarios. Technical Report (2008), <http://cis.cs.tu-berlin.de/Forschung/Projekte/bizycle/semca.pdf>
4. Boudjlida, N., Panetto, H.: Annotation of enterprise models for interoperability purposes. In: Proceedings of the IWAISE (2008)
5. Fabro, M.D.D., Bézivin, J., Jouault, F., Breton, E., Gueltas, G.: AMW: a generic model weaver. In: Proceedings of IDM 2005 (2005)
6. Hildenbrand, T., Gitzel, R.: A Taxonomy of Metamodel Hierarchies. University of Mannheim (2005)
7. Hoffmann, P.: Design of a model-based message transformation language. Diploma thesis, TU Berlin (2008)
8. Hohpe, G., Woolf, B.: Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley, Reading (2003)
9. InterSystems. Ensemble data transformation language (2006), <http://www.intersystems.com/ensemble/docs/4/PDFS/DataTransformationLanguage.pdf>
10. Kiryakov, A., Popov, B., Ognyanoff, D., Manov, D., Kirilov, A., Goranov, M.: Semantic annotation, indexing, and retrieval. In: Fensel, D., Sycara, K.P., Mylopoulos, J. (eds.) ISWC 2003. LNCS, vol. 2870, pp. 484–499. Springer, Heidelberg (2003)
11. Kühne, T.: Matters of (meta-)modeling. Software and System Modeling 5(4) (2006)
12. Kutsche, R., Milanovic, N. (Meta-)Models, Tools and Infrastructures for Business Application Integration. In: UNISCON 2008. Springer, Heidelberg (2008)
13. Kutsche, R., Milanovic, N., Bauhoff, G., Baum, T., Carlsburg, M., Kumpe, D., Widiker, J.: BIZYCLE: Model-based Interoperability Platform for Software and Data Integration. In: Proceedings of the MDTPi at ECMDA (2008)
14. Leicher, A.: Analysis of Compositional Conflicts in Component-Based Systems. Ph.D Dissertation, TU Berlin (September 2005)
15. Mehta, N., Medvidovic, N., Phadke, S.: Towards a Taxonomy of Software Connectors. In: Proceedings of the 22nd ICSE (2000)
16. Milanovic, N.: Contract-based Web Service Composition. HU Berlin (2006)
17. Milanovic, N., Kutsche, R., Baum, T., Carlsburg, M., Elmasgunes, H., Pohl, M., Widiker, J.: Model & Metamodel, Metadata and Document Repository for Software and Data Integration. In: Proceedings of the ACM/IEEE MODELS (2008)
18. Pulier, E., Taylor, H.: Understanding Enterprise SOA (2006) (manning)
19. Rahm, E., Bernstein, P.: A survey of approaches to automatic schema matching. VLDB Journal 10(4), 334–350 (2001)
20. Wache, H., Vögele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., Hübner, S.: Ontology-based integration of information - a survey of existing approaches. In: Proceedings of the IJCAI 2001 Workshop: Ontologies and Information Sharing, pp. 108–117 (2001)

Medical Personal Data in Secure Information Systems

Tatjana Welzer¹, Marko Hölbl¹, Marjan Družovec¹, Brane Klopčič²,
Boštjan Brumen¹, Hannu Jaakkola³, and Mirjam Bonačič¹

¹ Faculty of Electrical Engineering and computer Science, University of Maribor,
Smetanova ulica 17, SI - 2000 Maribor, Slovenia

{welzer,marko.holbl,marjan.druzovec,
bostjan.brumen,mirjam.bonacic}@uni-mb.si

² University Medical Centre Ljubljana, Department of Traumatology,
Zaloška cesta 2, Ljubljana

brane.klopacic@gmail.com

³ Tampere University of Technology, Pori, P.O. Box 300, FIN - 28101 Pori, Finland
hannu.jaakkola@tut.fi

Abstract. Between secure information systems (IS) are also medical IS which support work of different medical institutions as well as pharmacies and insurance companies. All of them have to work with medical personal data which should take into account the privacy. The privacy is the individual's right to determine if, when and how data about them will be collected, stored, used and shared with others. According to this definition medical personal data are treated as sensitive data, which can only be gathered and processed under particular conditions. In this contribution we will concentrate on personal medical data saved in medical records. Namely there are numerous message flows between medical staff and medical records that are often completely unprotected and can be accessed easier than might be expected. We will study the guidelines for medical staff regarding the protection of personal data, the corresponding Slovenian legislation, and the recommendations of a particular institution.

Keywords: Information systems, personal data, privacy, medical records.

1 Introduction

Information and information technologies are currently deployed on a broad scale in healthcare and medicine [15]. The fundamental use case is making medical information, based on medical data, quickly available for better and more efficient treatments of patients. An appropriate implementation guarantees for correct attribution of information and access control to guard privacy [15], [14], [16]. That means that security and the protection of data have become a very important part of everyday work and life in healthcare and medicine [16].

Privacy ensures that medical records are accessed only by authorized parties; only by those who should have access to something and are able to get access [10]. Access does not only include reading but also activities and possibilities that could be considered even more important: viewing, printing, auditing or simply knowing that a

particular record exists. Privacy aims to prevent the illegal or unauthorized intrusion into an individual's data (personal data) in every relevant area [9]. Personal data is data that defines an individual's property, state, or relationship, regardless of the format in which it is expressed. Personal data is combined into data medical records that are automatically or manually processed with the goal of faster and better treatments. Medical records are intended to be done within the limitations defined for any user of the data. Data is processed in the sense of collecting, storing, changing, uniting, deleting and transferring. The devices on which the data is stored are defined as devices on which data can be written or recorded. The user of medical records or rightful claimant is a subject who is by law or written consent authorized to establish, manage, maintain, supervise and use the medical record (data collection).

In this paper, we will introduce Slovenian legislation, and as a practical example, the regulations of the particular institution. We will conclude with some suggestions for a response to privacy needs in healthcare and medicine.

2 Data Protection in Slovenia

Data protection in Slovenia is restricted by contemporary legislative acts which tend to simulate real life situations, and mirror legislative acts from the European Union and similar guidelines [1], [7]. Slovenia has a modern Personal Data Protection Act. In the public sector, there are other laws and acts regulating the methods and time limits for the storage of documentation, which emerges in the process of administrative procedures and archive material. The acts define the physical, organizational and technical data protection methods as well as legal documents respective of internal provisions that companies and organizations have to develop for introducing and managing procedures and measures to protect data. The Healthcare Databases Act covers the medical records (medical data collections) that are managed, used and exchanged between legal and natural persons [13].

2.1 Personal Data Protection Act

Due to rapid technological progress, new ways of collecting, storing and processing personal data have arisen. Because of this, the need arose to renew the Personal Data Protection Act and at the same time implement European Union directives and guidelines [1], [7], [15] for the protection of individuals when processing their personal data and to assure the better free flow of such data. The Personal data Protection Act [2] in principle states that personal data protection is intended to prevent illegal and unauthorized interventions of one's private information. In this sense, the protection of personal data is not the protection of data as such, but rather the protection of one individual's rights, which the data refers to. Personal data can be processed only if so determined by law or if the manager behind data collection has the written consent of the individual. Legal and natural persons, who carry out a public service or a service restricted by the companies act, can on the basis of this act process personal data for people who are in a contractual relationship - but only if the personal data is needed for the fulfilment of contract obligations. For state agencies, local authorities and carriers of public mandates, the regulations are different as they can only process personal data

defined by law. For an individual whose personal data is being processed on the basis of their written consent, the consent has to be previously acquired in written form with the intention of data processing, the use and the period of storage [13].

2.2 Healthcare Databases Act

The Healthcare Database Act [3] covers data collecting for healthcare services, i.e. the collecting, process and transfer of data, which is carried out according to the law by legal and natural persons who provide health services. The assembling, processing and transfer of personal data included in data collecting is regulated by the personal data protection act, if not otherwise defined by this act. The data collection managers are from the Institute for Health Care of Slovenia and other healthcare providers in Slovenia. Data collection managers most often acquire personal data directly from individuals. They have the right to use health insurance numbers from health insurance cards as a linking point for collecting, processing and transferring data as defined by the law. The health insurance card infrastructure is also used for data collection, while simultaneously assuring separate access to health insurance data and health insurance card data. Data transferred between the health insurance card and other central collection agencies defined by this law are the ones needed for assuring health care services. Data collection via the health insurance card is managed by the Ministry of Health. Personal data can also be acquired by a doctor if this is necessary for the preservation of someone's life. If the personal data refers to the race, national or any other origin, political, religious or other belief or sexual orientation, the data collection manager can only acquire the data through an individual's consent [13].

3 Regulations for Personal Data and Other Sensitive Data Protection

On the base of previously described European guidelines [1], [7] and Slovenian legislation [2], [3], the particular institution manages a catalogue for data collecting, which describes all the relevant regulations [4]. Also the procedures and measures for protecting personal data (treated as a business secret) are regulated. In general, in different corporations and organizations, these kinds of catalogues define the organizational and technical procedures, and measures for the protection of confidential personal data to prevent access, processing, use, destruction, alteration and transfer of data.

The mentioned particular institution in its regulations defines under what circumstances the transfer of confidential data with the use of information technology is possible [4]. Defined is the transfer of physical media with confidential data and the procedure carried out on the receiver's side. For every transfer of personal and other confidential data the rightful claimant must apply in a written form and the application must be noted. Healthcare employees and other employees must protect any data that they access, and they do not have the right to communicate them to others with the sole exception of authorized personnel. The medical documentation includes all written data about patients, their diseases, family and other relations. The first copies are stored in the institute and are protected. The institution has to enable patients' access to their medical records and a copy of their medical records within the defined time limit.

The medical doctor treating the patient must make a judgment, based on the patient's state of health, which data can be revealed to the patient without causing harm or deterioration to their health [5]. The patient decides how much information can be given to their relatives and the public about their health state. Healthcare personnel can be released from their obligation to professional secrecy by the patient or by a court, with the exception of cases where it is deemed that secrecy is to the benefit of the patient. After a patient's death, specific members of the family can grant access to the diagnosis and epicrisis of the patient on behalf of a written claim, except in cases when the deceased has strictly prohibited it. The data can be used from the institution also for statistical purposes or medical research work, but only in a form that does not enable the identification of a patient [13].

The institution can process data by using prescribed software and software in accordance with the standards. Software has to be licensed with a license that permits the installation and use of programs at a fixed number of locations.

User identification and authorization system enable access to data and the use of other sources. For access attempts, a log is maintained which is managed by an employee competent in data protection. The time limit for accessing data storage is equal to the time limit for protected records. After the time limit passes, the data has to be archived. When data access and records of access to protected data is based on passwords, the user must change their password after three months at the latest. The changed password must not be the same as the old one.

All software and data stored on the main computer of the institution must have a copy and a second copy. The second copy has to be stored at a safe location outside the building of the main computer. Access to the software must be protected in such a way that access is granted only to specific employees or legal/natural persons. Altering the system or application software is only allowed with an authorized person's consent. The number of people who have access to personal and other confidential data has to be as low as possible.

After the time period defined by the appropriate law or regulation, data has to be deleted from the collection. The time periods for deletion are defined in the data catalogue. For deleting computer data, methods are applied that prevent the restoration of all or part of the data. The deletion of data must be carried out by a commission and minutes must be taken.

4 Conclusion

To protect personal medical data we have gotten an overview of the knowledge demands on information privacy for personal data in medical records, which also govern the regulations of our selected institution [1], [2], [3], [4], [7].

Presented topics are very sensitive and comprehensive. They demand continuous work on regulations, recommendations and related reports [5]. It is important that healthcare and medical personnel are aware of security, privacy and their own responsibility for it [8]. They also need to be able to give proper and/or adequate information to patients. Receiving spoken information is for the patient not enough (may be they can not understand it in the spoken form, they need more time to process information like this,...). From mentioned reasons, they should get also written material and training

(informative meetings, courses, workshops) with the goal of making them aware of the importance of informational privacy. Patients need to be aware of their own rights, as well as the rights of relatives and/or legal representatives connected with their medical records.

Also healthcare and medical personnel have to be educated. We suggest introducing appropriate training courses for healthcare and medical personnel on relevant topics, as well as establishing educational programs as a response to information technology's influence on healthcare and medical systems [8]. Training can be organised either informally or on an internal level by information technology staff or formally by companies and different institutions that are specialised in information privacy, including information privacy for medical records. Permanent education can be carried out by formal educational institutions like universities, faculties, and schools with educational programs that are specialised in information technology for healthcare and medicine. Programs can be either undergraduate or postgraduate. According to our experience, the latter provide better results [6] since participating students already have knowledge and experience in healthcare and can therefore cope more easily with the new topic, even though many are surprised (if not shocked) as to why they should study a specialised topic (general security, privacy) that is so far afield from their main topic of study – healthcare or medicine [12]. On the other hand, undergraduate students are generally not handicapped by primary and secondary topics, since they are younger, and therefore often have more information technology backgrounds as postgraduate students.

Information privacy for personal data in medical records is a very important topic, which is not secured by guidelines, legislation and regulations. What is much more critical is that everyone: personnel, patients, relatives, legal representatives and other participants, who needs to take part in producing, using, storing, demanding, etc. data for medical records has to be aware of the importance of information privacy.

To be able to provide the community with more official judgement and data, some research have to be done in different institutions that has to be organised as our particular institution. It would be also very important to build-in, in secure information systems, all gained knowledge and experiences, all with the goal for more secure personal medical data, observing the security from different points of view from the patient to his/her relatives as well as healthcare and medical personal.

References

1. European Guideline for Medical Personnel on protecting Personal Data (2005) (accessed, June 2008), <http://www.eurosocap.org/>
2. Personal Data Protection Act. Official Journal of the Republic of Slovenia, No. 94/2007 (2007)
3. Healthcare Database Act. Official Journal of the Republic of Slovenia, No. 65/2007 (2007)
4. Regulations for personal data and other sensitive data protection and documented material of the Medical Centre (available only in Slovenian), Medical Centre Ljubljana (2006)
5. Klemenc, D., Požun, P., Milič, J.: Privacy of the patient's personal and medical data in the University Medical Centre Ljubljana. *Informatika Medica Slovenica* 9(1-2), 24–30 (2004)

6. Welzer, T., et al.: Teaching IT in the postgraduate health care and nursing program; Advancing health information management and health informatics: issues, strategies, and tools. In: Raza, A., Bath, P., Keselj, V. (eds.) Eleventh international symposium on health information management research - iSHMIR 2006, pp. 14–16 (2006)
7. SEISMED Consortium (ed.): Data Security in Health Care, Guidelines. IOS Press, Amsterdam (1996)
8. Yu, H., Liao, W., Yuan, X., Xu, J.: Teaching a web security course to practice information assurance. *ACM SIGCSE Bulletin* 38(1), 12–16 (2006)
9. Pfleeger, C.P., Pfleeger, L.: Security in Computing. Prentice Hall, Englewood Cliffs (2007)
10. Cannon, J.C.: Privacy. Addison-Wesley, Reading (2005)
11. Kokol, P., Zazula, D., Brumec, V., Kolenc, L., Slajmer Japelj, M.: New Nursing Informatics Curriculum - An Outcome from the Nice Project. In: Mantas, J. (ed.) Proceedings of HTE 1998, University of Athens (1998)
12. Welzer Družovec, T., Hölbl, M., Habjanič, A., Brumen, B., Družovec, M.: Teaching of Information Security in the Health Care and Nursing Postgraduate program. In: Venter, H. (ed.) IFIP TC-11 International Information Security Conference - SEC 2007, IFIP International Federation for Information Processing, vol. 232, pp. 479–484 (2007)
13. Welzer, T., et al.: Information privacy for personal data in medical records. In: Bath, P. (ed.) ISHIMR 2008: Proceedings of the Thirteenth International Symposium for Health Information Management Research, October 20–22, 2008, pp. 149–157. Massey University, Auckland (2008)
14. Joosten, R., Whitehouse, D., Doquenoy, P.: Putting Identifiers in the Context of eHealth. In: Fischer-Hübner, S., Doquenoy, P., Zuccato, A., Martucci, L. (eds.) IFIP International Federation for Information Processing. The Future of Identity in the Information Society, vol. 262, pp. 389–403. Springer, Heidelberg (2008)
15. i2Health – Interoperability Initiative for a European eHealth Area – project deliverable D3.1b Identification management in eHealth (2007) (accessed, June 2008), <http://www.i2-health.org/>
16. Yee, G., Korba, L., Song, R.: Ensuring Privacy for E-Health Services. In: Proceedings of the First International Conference on Availability, Reliability and Security - ARES 2006, pp. 321–328. IEEE Press, Washington (2006)

Implementing Medication Management Software Effectively Within a Hospital Environment: Gaining Benefits from Metaphorical Design

Salah Awami¹, Paula M.C. Swatman¹, and Jean-Pierre Calabretto²

¹ School of Computer and Information Science, University of South Australia
City West Campus, Adelaide, SA, 5000, Australia

Salah.Awami@postgrads.unisa.edu.au,

Paula.Swatman@unisa.edu.au

² School of Pharmacy and Medical Sciences, University of South Australia
City East Campus, Adelaide, SA, 5000, Australia

Jean-Pierre.Calabretto@unisa.edu.au

Abstract. Implementing health information systems (HIS) to support the healthcare process is subject to many challenges: behavioural, technical, and organisational. Developing these technological artefacts based on good understanding of such challenges, yields a system design capable of addressing implementation issues. Based on such understanding, we set off a development process to produce a metaphoric software tool to support health care practitioners in a hospital setting. The development of software adopting such approach explicitly considers the inclusion of users concerns, a crucial determinant for the successful implementation of any IS. The software design directly implemented the look and feel of a paper-based medical form used by targeted health care practitioners. This paper illustrates empirical research done in developing and evaluating a metaphoric software tool, and highlights important aspects of such approach in addressing the implementation process.

Keywords: Implementation of Health Information systems, metaphoric design, socio-technical approach.

1 Introduction

The healthcare environment is an increasingly complex area characterised by advances in medical science and technology, growing specialisation, ever-greater patient expectations and, most critically of all, by the size and variety of healthcare itself [1]. The interaction space within which health practitioners work is also complex in terms of the many social networks across which individuals must communicate [2]. One major way of managing these complexities is through the use of Information Systems (IS).

Despite their obvious importance, however, IS implementations within the healthcare environment can fail for a variety of reasons – technical, social or behavioural – including:

- System attributes: such as (perceived) ease of use, ease of learning; and additional work required [3-6].
- User resistance: including factors such as interference with the healthcare practitioner's workflow, and lack of time to deal with the system [7]. The changes associated with IS implementation are often resisted by users when they involve some type of loss to the users – loss of time required for training, loss of control; or even loss of routine work practices [8].
- Conception-reality gaps: which occur when health care managers adopt leading-edge technologies without realising the magnitude of the change they will impose. Heeks et al. [9] refer to cutting-edge IS as reality-supporting or rationality-imposing applications. Reality-support applications closely mimic existing realities, providing features such as free text-based recording, pen and audio input, and mobile computing [9]. They recommend adopting IS which impose only limited changes to current business processes, highlighting the importance of incremental IS implementation approaches, rather than imposing radical changes.
- Techno-centric implementation: a frequent issue for medical IS implementation, where the process is seen as a purely technical issue without consideration of the social and organisational aspects of the environment – as recommended by the socio-technical approach [10].

While no single approach to the design of IS can ever hope to resolve all these issues, there are ways of reducing the failure rate of hospital-based IS and increasing their acceptance by healthcare workers. Metaphorical design [11-14], which adapts an object from users' work processes to act as a metaphor for the design of the software user interface, focuses on providing users with an easily comprehensible interface and working environment – and thus minimising the difficulty of adjusting to and learning any new software system. The project reported in this paper takes a metaphorical approach to the design of a software tool to support pharmacists and other healthcare practitioners (predominantly doctors and nurses on the ward) involved in managing medication in an Australian acute care hospital. We hope that this will address some of the human behavioural issues associated with implementing a software tool – as well as assisting in understanding the challenges of IS usability, users' conception-reality gap; and the social aspects of the IS implementation process.

The next section of this paper is a review of the literature which explains the concept of a metaphor, its origin and how the metaphoric approach might benefit software design, followed by a brief description of the research methodology. The general approach adopted in the research is discussed in more detail in the section which follows. Background information about the targeted process and the software context in the hospital is then outlined and conclusions are drawn in the final section.

2 Literature Review

Metaphors are widely used in human expression and thoughts. When people use metaphors, they are normally trying to explain one concept, new to the recipient, by means of another concept which is familiar and readily understandable. In the software development context, metaphors are often used when implementing application

interfaces with the goal of improving software usability [15] because they can exploit users' prior experience and knowledge from other domains to learn about novel technological artefacts [14]. The concept of adopting objects from real life to facilitate users' work on computers was most famously applied in developing the modern desktop metaphor, whose origin remains a matter of dispute between Xerox, Apple and IBM [12].

Metaphors are not only restricted to facilitating users' experience with the IS, but can also play a role in generating good design ideas about the entire system, whether visible to the user or not [14]. Metaphors can be implemented either visually or conceptually. Blackwell [12] points out that their purpose is to narrow the gap between computer systems and the users' real world; noting the need for developers to consider the human dimension(s) of the artefact – a crucial requirement for producing successful information systems.

Naturally, software developers begin by thoroughly investigating the domain which is the target for software support. Reaching a level of understanding of the problem domain, enough to stimulate a technical solution, is an important domain investigation outcome [16, 17]. This crucial phase of the system development process requires active knowledge acquisition and integration among the development team members, with the goal of collectively forming a shared understanding of the problem and potential solution [18, 19]. Software designers generate a solution design based on their understanding of the problem domain which emerges from the analysis phase. The same sequence of development tasks applies in developing the software user interface: best practice design guidelines suggest adopting metaphors to inform software interface design [20].

A rather different perspective on our situation (addressing the implementation of a software tool in a hospital setting) is provided by Berg and Toussaint [21]. These authors remark, on the basis of a socio-technical approach and a deep understanding of the healthcare system, that typical system development approaches might not produce the optimal IT solution to support healthcare processes. They suggest developing information systems by reengineering an important object used by healthcare practitioners: paper-based medical records. Berg and Toussaint [21] criticise current modelling practice where the aim is to model an interactive contingent process (the healthcare process), believing that this is almost impossible. They propose, instead, investigating the paper-based medical record because it models 'a bit of everything out of the field of work in which it operates' [21, p7] and learning from its strengths and successful implementation story to formulate models for electronically developed systems.

A final perspective informing our analysis emphasises adopting simple supporting technology tools, rather than a complex system, as an intervention to support work process in a social context such as a hospital environment; and ensuring that the decision support provided was not so complex as to interfere with user activities [22].

3 Methodology (Design Research)

We chose to adopt Design Research as our methodology because it is flexible, iterative and target-oriented [23, 24]. Target orientation is of particular relevance because our research – in addition to contributing to existing knowledge on human factor

influence in software implementation process in hospitals – is also directed toward creating a working information system to support health care practitioners. Design research targets the development of a solution with the desired functionality, which performs according to expectations [23].

Design research methodology is usually conducted on a well-understood series of steps [25]. It begins with the analysis step where the problem domain is analysed, followed by a design phase in which solution specification is suggested. The construction phase then creates a system based on the suggested design. Finally, the implementation and evaluation phases investigate artefact's usefulness and conformance to expectations – marking the end of one cycle. The outcome from the last step in the cycle feeds into the analysis step of another cycle of system development. At the end of each cycle an improved version of the artefact is achieved. This iterative approach provides a means for exploring better ways of addressing IS design issues and allows feedback to further refine the expected outcome. The approach also enables a better understanding of the problem domain targeted by the analysis.

Design research yields artefacts which can include a construct, a model, a method, or an instantiation – although this ordering does not suggest any sequential order between the artefacts. In fact, the creation of models or constructs may occur after the creation of an instantiation: the system itself [23], as Figure 1 shows in a slightly different way.

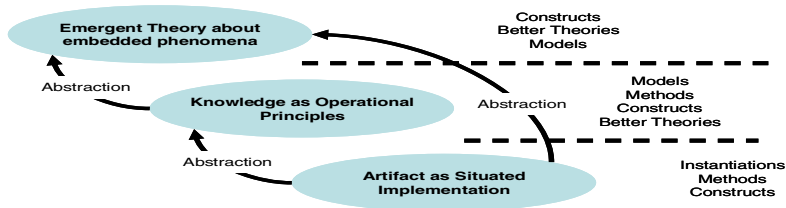


Fig. 1. Outputs of Design Research [25]

We have currently completed one cycle of this research project. Within this cycle we have performed analysis, chosen a design, implemented an early version of our software; and undertaken user evaluations.

4 Research Approach and Context

A major factor in the development of our software tool was the human dimension of the medication management system, which we addressed by adopting a holistic perspective of the needs of the varied group of health care practitioners who would be using the software tool. The fact that this tool must work effectively within a complex socio-technical system dictated that development efforts proceed in three directions simultaneously:

- firstly, on the behavioural front, we reviewed the information systems literature and developed an understanding of the behavioural challenges affecting HIS implementation

The candidate object adopted from the users' work practice to act as a software metaphor was a medication-related form already being used by the targeted practitioners (pharmacists, medical officers and nurses): the Medication Management Plan (MedMAP) (Figure 2).

The first version of the paper form had originally been known as the MATH (Medication & Allergy Tracking History), but had then undergone repeated development rounds leading to its current format: the MedMAP form. Research carried out on the form to evaluate its impact suggested that it provided a satisfactory tool for effective management of medication information.

Other reasons for selecting MedMAP as the metaphor source included its recent development and wide use by the relevant group of healthcare practitioners within a number of South Australian hospitals, which suggested it would provide the necessary enhancement effect on work existing processes. Now we needed to exploit information and communication technology (ICT) capabilities to further empower these same processes.

Fig. 2. The paper-based MedMAP form – the metaphor source

We wanted a simple intervention which would have the least intrusive impact on users' existing work routines, so as to minimise user resistance, as our approach would be considered low in terms of 'loss' from the user perspective. The electronic version of MedMAP (eMedMAP) which we developed is shown in Figure 3.

The screenshot displays the eMedMAP software interface. At the top, there's a header with the 'eMedMAP' logo and navigation buttons like 'Front page' and 'Back Page'. Below this, patient details are entered: Family name (Kalen), Given name (Bruce), Address (123 some street), Date of birth (12/06/1958), Medicare no (K822222), Height (1.70), Weight (78), and Concession (CN11205). A section for 'Allergies & Adverse Drug Reactions' includes a table with columns for Drug (or other), Reaction, Date, and Status. Below this is a 'Patient Requirements on Discharge' section. The main part of the screen is a table listing medications with columns for Reg, Generic name, Dose, Freq, Route, Comments, Started, Ceased, By, and At home Supply. A dropdown menu is open, showing a list of medications including GENTAMICIN SULFATE, GEMFIBROZIL, and GEMFIBROZIL 600mg TABLETS. At the bottom, there are sections for 'Source of Medication History' and 'Prescription Prior to Admission'.

Fig. 3. The electronic version of MedMAP (eMedMAP)

5 Findings and Discussion

To evaluate the impact of adopting a metaphorical system design on users' reaction to the system, we needed to undertake a significant evaluation process: including investigating users' opinion about using the metaphor-based eMedMAP tool; and how successful the software was in mimicking the look and feel of the original paper-based form. The evaluation also solicited feedback on the software tool, with the goal of eliciting additional user requirements which could be implemented in later versions of the software.

A total of eight pharmacists tested the software in thirty-minute evaluation sessions. They were asked to fill in an evaluation form to reflect on their perception of eMedMAP and this analysis provided the following outcomes:

- All participants agreed that the underlying concept of a software tool with the look and feel of a familiar paper-based medical form already in use was a useful concept.
- Other aspects of eMedMAP investigated led to the following responses:

- All participants stated that eMedMAP matched its paper-based counterpart.
- All participants confirmed they did not need to attend any training in order to use the software – and 5 of the 8 pharmacists interviewed stated they needed no help at all to use the software.
- 5 of the participants believed they were familiar with the contents of eMedMAP.
- All participants agreed that the flow of tasks in eMedMAP was natural and matched their normal daily task activities.
- 3 of the participants agreed that the software would fit naturally into their work practice, with a further 2 asking about software portability which is considered crucial to the adoption of eMedMAP.

In relation to the question ‘will basing the software design on a metaphor provide a common understanding among users?’ 62.5% of participants agreed that the software functionality met their expectations, even though not all participants had participated directly in the software development effort.

6 Limitations and Challenges

There were a number of limitations which could affect the uptake and successful adoption of eMedMAP:

- Physical access to the software tool in the wards: the software will need to be available on a tablet PC (or PDA) to ensure continuity with the original paper-based form and preserve existing work practices.
- Procedural limitations: the intervention we developed was intentionally limited to support a particular process. Extending the capabilities of this software solution to include connecting to medical evidence-based information to support decision making is a separate process. However, integration with other information systems within the hospital is the objective of further refinements of the software.

7 Conclusions

This paper investigates a particular approach to addressing the implementation challenges of a software tool within a hospital environment. Based on a holistic perspective, the development process of this software tool aimed at producing an effective design capable of addressing the foreseen implementation issues. Specifically, we used a metaphoric design approach to ensure the inclusion of the human element within a socio-technical system. Understanding users’ concerns and including these in the system design potentially yields a highly implementable system. Metaphoric system design ensures this aspect by adopting an object from the users’ own work practice to inform system UI design and act as a system model. Adopting such an approach ensures that the design produced is relevant to the targeted users and processes.

This research is also motivated by our belief that, in a healthcare-related context such as a hospital, any intervention should be user-centred and provide a simple

support tool for professionals such as health care practitioners who value their autonomy and independence. Adopting the users' own perspectives helps to establish their ownership of the new system [10] and motivate them to participate actively in the IS implementation efforts. This, we believe, will assist in addressing the 'loss of control' concern often articulated by user groups.

Briefly, we argue that using an object from the users' work practice as a system metaphor has the following benefits:

- An effective facilitator for the discussion happening at the start of the software development process between designers and users. Such dialogue helps to develop the necessary mutual understanding of the process targeted by the intervention, which is normally outside the expertise domain of the software developer.
- Offers a rich insight into the targeted process for the system developer.
- Provides a real indicator of users' involvement in the system design – an essential prerequisite for successful implementation of the software.
- Ensures that the system model developed is user-oriented rather than being a model developed from the designers' perspective, supporting users' expectations about what can be done and how the system works.
- Created a tool to generate effective design options for the developing system.
- Potentially reduces the gap between users' workplace realities and new, system-required tasks, increasing the likelihood that the new solution will fit naturally into users' work processes.

References

1. Pope, C., Mays, N.: Qualitative Research: Reaching the parts other methods cannot reach: an introduction to qualitative methods in health and health services research. *British Medical Journal* 311(6996), 42–45 (1995)
2. Ash, J.S., Berg, M., Coiera, E.: Some Unintended Consequences of Information Technology in Health Care: The Nature of Patient Care Information System-related Errors. *Journal of the American Medical Informatics Association* 11(2), 104–112 (2004)
3. Van der Meijden, M.J., et al.: Determinants of Success of Inpatient Clinical Information Systems: A Literature Review. *J. Am. Med. Inform. Assoc.* 10(3), 235–243 (2003)
4. Southon, F.C.G., Sauer, C., Dampney, C.N.G.: Information Technology in Complex Health Services: Organizational Impediments to Successful Technology Transfer and Diffusion. *J. Am. Med. Inform. Assoc.* 4(2), 112–124 (1997)
5. Davis, G.B., et al.: Diagnosis of an information system failure: A framework and interpretive process. *Information & Management* 23(5), 293–318 (1992)
6. Gerlach, J.H., Kuo, F.-Y.: Understanding Human-Computer Interaction for Information Systems Design. *MIS Quarterly* 15(4), 527–549 (1991)
7. Ash, J.S., Bates, D.W.: Factors and Forces Affecting EHR System Adoption: Report of a 2004 ACMI Discussion. *J. Am. Med. Inform. Assoc.* 12(1), 8–12 (2005)
8. Lorenzi, N.M., Riley, R.T.: Managing Change: An Overview. *J. Am. Med. Inform. Assoc.* 7, 116–124 (2000)

9. Heeks, R., Mundy, D., Salazar, A.: Understanding Success and Failure of Health Care Information Systems. In: Armoni, A. (ed.) *Healthcare Information Systems: Challenges of the New Millennium*. Idea Group Publishing, London (2000)
10. Berg, M.: Patient Care Information Systems and Health Care Work: a Sociotechnical Approach. *International Journal of Medical Informatics* 55(2), 87–101 (1999)
11. Madsen, K.H.: A guide to Metaphorical Design. *Commun. ACM* 37(12), 57–62 (1994)
12. Blackwell, A.F.: The reification of metaphor as a design tool. *ACM Trans. Comput. -Hum. Interact.* 13(4), 490–530 (2006)
13. Noble, J., Biddle, R., Tempero, E.: Metaphor and metonymy in object-oriented design patterns. *Aust. Comput. Sci. Commun.* 24(1), 187–195 (2002)
14. MacLean, A., et al.: Reaching through analogy: a Design Rationale perspective on roles of analogy. In: *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*. ACM, New Orleans (1991)
15. Erickson, T.D.: Working with interface metaphors, in *Human-computer interaction: toward the year 2000*, pp. 147–151. Morgan Kaufmann Publishers Inc., San Francisco (1995)
16. Kinzie, M.B., et al.: A User-centered Model for Web Site Design: Needs Assessment, User Interface Design, and Rapid Prototyping. *J. Am. Med. Inform. Assoc.* 9(4), 320–330 (2002)
17. Walz, D.B., Elam, J., Curtis, B.: Inside a software design team: knowledge acquisition, sharing, and integration. *Commun. ACM* 36(10), 63–77 (1993)
18. Kensing, F., Blomberg, J.: Participatory Design: Issues and Concerns. *Computer Supported Cooperative Work (CSCW)* 7(3), 167–185 (1998)
19. Holtzblatt, K., Beyer, H.: Making customer-centered design work for teams. *Commun. ACM* 36(10), 92–103 (1993)
20. Weinschenk, S., Jamar, P., Yeo, S.C.: *GUI Design Essentials*. John Wiley & Sons, Inc., Chichester (1997)
21. Berg, M., Toussaint, P.: The mantra of modeling and the forgotten powers of paper: a sociotechnical view on the development of process-oriented ICT in health care. *International Journal of Medical Informatics* 69(2-3), 223–234 (2003)
22. Calabretto, J.-P.: Supporting Medication-related Decision Making with Information model-based Digital Documents. In: *School of Computer and Information Science*. University of South Australia, Adelaide (2007)
23. Hevner, A.R., et al.: Design Science in Information Systems Research (1) (Research Essay) 28(1), 75(31) (2004)
24. March, S.T., Smith, G.F.: Design and natural science research on information technology. *Decision Support Systems* 15(4), 251–266 (1995)
25. Vaishnavi, V., Kuechler, B.: Design Research in Information Systems, January 18 (2006) (cited July 15, 2007),
<http://www.isworld.org/Researchdesign/drisISworld.htm>

Exploring the Potential of Component-Oriented Software Development Application

Hazleen Aris

Universiti Tenaga Nasional, 43009 Kajang, Selangor, Malaysia
hazleen@uniten.edu.my

Abstract. This article presents about a study performed to investigate the current state of COSD application amongst software developers in Malaysia. The information required for the study was obtained through questionnaires that were distributed to the software developers who are working at various software development companies in Malaysia. Results from the study are used to determine the potential of COSD application in Malaysia. Outcomes of the analysis performed on the results show that the potential of COSD application amongst the software developers in Malaysia is high.

Keywords: COSD practice, COSD survey, software development practice.

1 Introduction

The idea of component-oriented software development (COSD) is as old as the software engineering itself with its inception dates back to the late 1960's. Since then, it has become one of the central research topics in the field of software engineering, and its evolution has taken place in many different forms and from various aspects. The main motivation behind these research is, of course, the list of advantages expected to be brought about by the successful implementation of COSD and the success stories of components reuse from other engineering fields, particularly in mechanical and civil engineering. Despite the presence of a number of obstacles along the path in adopting COSD, which are hindering its application from taking off with a blast, researchers are still optimistic that these obstacles, or rather, challenges can be 'eliminated' as COSD matures [1]. Hence, research on COSD is becoming more and more intense with each addressing one or more challenges imposed on COSD.

One of the areas that have caught the attention of researchers in this field is the study on the current practice amongst the software developers, software companies and software projects, with regard to COSD. Motivated by the work done by [2] that evaluated the current state of COSD in the Kingston city of Jamaica, we performed a similar study to find out the current state of COSD application amongst the software developers in Malaysia. Our purpose is however slightly different. From this study, we hope to be able to determine the potential of COSD application amongst the software developers in Malaysia.

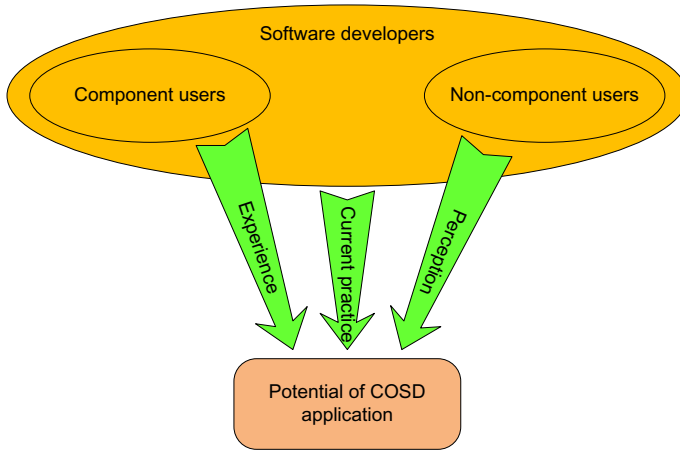


Fig. 1. Reference model for information gathering

To achieve the purpose, we formulated a model shown in Fig. 1 to help us in identifying the sources and types of information to be collected that will help us in determining the potential of COSD application. As can be seen from the model, the software developers are divided into two categories; component users and non-component users. Component users are the software developers who admit to using components in their software development projects. Conversely, non-component users are the software developers who do not use components in their software development projects. From the component users, we would like to obtain information on their experience in using components in order to determine the potential of COSD application. From the non-component users, we would like to know their perception towards COSD in order to determine the potential of COSD application. From both component users and non-component users, information on their current practice in software development would be able to help us in determining the potential of COSD application.

2 Information Gathering

To gather the required information, questionnaires entitled ‘A Survey to Investigate the Current State of the Application of Component-oriented Software Development (COSD) amongst the Software Developers in Malaysia’, were distributed to the software developers working at various software development companies in Malaysia. The choice of survey method through questionnaire distribution for the study is in line with the recommendation made by Lethbridge et al. [3], due to its suitability to cater our target *sampling frame* [1], i.e. software developers who are working at various software development companies located at diverse geographical locations in Malaysia.

¹ Sampling terminology used according to Conradi [14].

Table 1. GQM table for the questionnaire

Goal	Purpose Issue Object (process) Viewpoint	Identify the potential of COSD application from the software developers' (Malaysia) viewpoint
Question	Q1	What are the (programming) languages used in the software development projects?
Metric	M1	% of each programming language used
Question	Q2	What is the methodology applied in developing the software?
Metric	M2	% of each methodology used
Question	Q3	What is the amount of codes reused (in percentage) from the previous projects?
Metric	M3	Mod of % of amount of codes reused
	M4	$\frac{\text{Reusers}}{\text{Total respondents}} \times 100$
	M5	$\frac{\text{Reusers of } > 50\%}{\text{Total reusers}} \times 100$
Question	Q4	Are you familiar with the term component with respect to software development?
Metric	M6	% of those who are familiar
Question	Q5	Do you use components in developing software?
Metric	M7	% of those who use components
Question	Q6 a) i.	Based on your experience, does the use of components reduce the number of errors found in the software?
	Q6 a) ii.	Based on your experience, does the use of components reduce the time taken to test the software?
	Q6 a) iii.	Based on your experience, does the use of components reduce the number of complaints from the customers/users?
Metric	M8	% of component users who experience all Q6 a) i) to Q6 a) iii
Question	Q6 b)	Based on your experience, does the use of components reduce the time taken to market the software?
	Q6 c)	Based on your experience, does the use of components reduce the cost of producing the software?
Metric	M9	% of component users who experience all Q6 a) to Q6 c)
Question	Q7	Despite the problems faced, do you still believe that COSD is a better way to develop software?
Metric	M10	% of component users who believe so
Question	Q8	Given the chance, will you apply component in your future software development?
Metric	M11	% of non-component users who will apply
Question	Q9	Based on your knowledge about components and/or the description about component given, do you think that software development by using and reusing components is a better way to develop software?
Metric	M12	% of non-component users who agree

There were a total of 19 questions in the questionnaire. Table 1 however only lists down questions that are relevant to the scope of discussion of this paper, i.e. determining the potential of COSD application in Malaysia. Table 1 also shows the corresponding metrics used to evaluate the questions in accordance to the goal-question-metric (GQM) paradigm [4]. A question may need to be divided into a number of subquestions to facilitate the respondents in understanding and answering the question. Subquestions of a question are listed alphabetically such as Q6 a), Q6 b) et cetera. Subsubquestions, where needed, are numbered with i., ii. and so on.

2.1 Questionnaire Design

To make the questionnaire more organised and more structured, it was divided into 4 main sections; section A, section B, section C and section D. Section A contains questions that ask for the background (demographic) information of the respondents. Section B contains questions that find out the respondents' current practice in developing software. The last question in section B asks about whether or not the respondents are using components in developing software. Based on the respondents' answers to this question, they would have to either proceed to section C, which is meant for the component users or section D for the non-component users. Section C contains questions that are looking for information on the nature of components used, problems in using them and ways to improve their use. Finally, section D contains questions that find out the perception of non-component users towards COSD.

2.2 Population Determination

The sampling frame for the questionnaire distribution is the software developers working in the Multimedia Super Corridor (MSC) status companies, which are clustered mainly at four cybercities in Malaysia; Kuala Lumpur city centre, Technology Park Malaysia, UPM-MTDC and Cyberjaya [5]. Our main source of information to estimate the number of software development companies in Malaysia is the MSC portal [6] where a list of information and communication technology (ICT) related companies with the MSC status from a number of sectors is made publicly available. At present², there is a total of 1,511 ICT related companies being granted the MSC status as shown according to their respective sectors in Table 2. This portal is found to be the most reliable source of information in estimating the number of software development companies in Malaysia and referred by a number of research in related area [5], [7], [8], [9]. Furthermore, the list available from the MSC portal is updated regularly and complete information on each company, including their addresses and contact numbers are made available for public view.

From the six sectors listed in the MSC portal, one particular sector is identified to be directly involved in the software development activities, which is the software development sector. With reference to Table 2, it can be said that our sampling frame is 803 software developers from the companies belonging to this sector. However, it is worth to note here that, as of November 2006, only 81.20% of the companies from the software development sector are still active and conducting MSC approved activities as discovered by [9]. This reduces our sampling frame to 652 software developers from these companies. From the 652 software development companies, we randomly selected 400 companies to distribute the questionnaires (sample).

Two forms of questionnaire were distributed; paper-based and web-based. Paper-based questionnaires were either posted or faxed to the companies. Web-based questionnaire participants were invited through emails. The online version

² True as of 13th April 2007.

Table 2. ICT related companies with MSC status

Sector	Number of Companies
Software development	803
Creative multimedia	149
Support services	143
Internet-based business	183
Hardware design	133
Shared services and Outsourcing	100

was created with the hope of increasing the response rate, as a study showed that the response rate using web-based questionnaire is 50% higher than the response rate using paper-based questionnaire [10]. Furthermore, the web-based questionnaire has additional features such as basic result analysis and skipping pattern that reduces possible human error in answering the questions.

3 Data Presentation

The information gathering exercise was eventually concluded in October 2007. At the end of the information gathering exercise, a total of 104 responses were received (subsample), making up 26% response rate. Some of the questionnaires were not able to be distributed due to:

- invalid addresses—the address published in the MSC portal is no longer used and so do the contact numbers
- overseas addresses—the development work for the companies are actually done overseas and
- branch company—the companies are branches of larger companies and therefore we only sent questionnaires to the parent companies.

The main reason given by those who refused to participate was time constraint. Other than the time constraint, quite a number of the companies' representatives contacted during follow up to non-responses said that they do not have any software developers in the companies i.e. they are just software resellers. Even though the response rate was only 26%, we decided to proceed with analysing the responses. Furthermore, according to Lethbridge et al. [3], for an exploratory study of this kind, a low response rate of about 5% would already be sufficient. The next section therefore presents the analysis performed on the information gathered **from 101 valid responses**.

3.1 Current Practice

As described in subsection 2.1, section B of the questionnaire aimed at obtaining information on the current practice of the software developers. Information on the portion of the software developers who are using components in their

Table 3. Programming languages used by developers (Q1)

Languages	Frequency	Percentage
C	26	25.74%
C++	28	27.72%
Java	42	41.58%
Visual Basic	41	40.59%
Others	56	55.45%

Table 4. Methodology applied in software development (Q2)

Methodology	Frequency	Percentage
Conventional	53	52.48%
Object-oriented	66	65.35%
Component-oriented	30	29.70%
Others	9	8.91%

Table 5. Amount of codes reused from previous projects (Q3)

Reuse amount	Frequency	Percentage
100%	5	4.95%
Around 80%	22	21.78%
Around 50%	49	48.51%
Less than 20%	20	19.80%
None	5	4.95%

software development projects is also sought. Table 3 shows the distribution of the programming languages used by the respondents. It can be seen that the most commonly used languages are Java (41.58%) and Visual Basic (40.59%). ‘Others’ languages used by the respondents range from COBOL to a complete development framework. From Table 4, it can be seen that the methodology applied, as claimed by most respondents is object-oriented (65.35%), followed by conventional methodology (52.48%) in the second place and component-oriented (29.70%) in the third.

Table 5 shows that 95.05% of the respondents ever reuse codes from the previous software development projects with the usage amount ranging from less than 20% to 100%. The mod of reuse amount is around 50% and the percentage of software developers who reuse around 50% and above of the codes from previous projects is 79.17%.

When asked whether or not they are familiar with the term component, 65 (64.36%) of the respondents are familiar with it and the balance of 36 (35.64%) respondents are not, as shown in Fig. 2. From the total of 101 respondents, 58 (57.43%) of them use components in their software development projects as shown in Fig. 3.

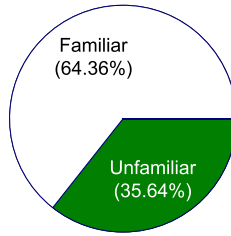


Fig. 2. Familiarity of the term component (Q4)

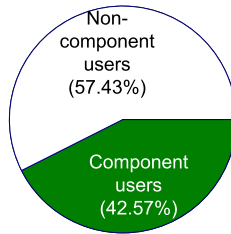


Fig. 3. Percentage of component users and non-component users (Q5)

3.2 Component Users' Experience

The advantages of COSD as stated in many literatures are higher software product quality, lower production time and lower production cost. For those who have been applying COSD in their software development projects, these benefits may have been materialised. For others, they remain a theory. Our interest here is to find out the percentage of component users who have experienced these advantages. Quality is further refined to comprise three contributing attributes; number of errors found, testing time and number of complaints from users. Component users are regarded as to have seen the increase in quality of the software produced when they agree to all of these attributes, as shown shaded in the Venn diagram of Fig. 4. Therefore, 39 (67.24%) component users actually see or experience an increase in the quality of the software produced when using components.

Out of these 39 component users, more than half of them (74.36%) also agree to the reduced time to market the software and lower cost in producing them as shown in the Venn diagram of Fig. 5. As such, it can be concluded that a total of 29 (50.00%) component users actually see the oft-mentioned advantages of COSD based on their experience.

In the questionnaire, questions about the problems faced in applying COSD were also asked to the component users. Various reasons were given and discussed in [11]. Interestingly, despite all the problems faced by these component users, 55 (94.83%) out of 58 of them still believe that COSD is a better way to develop software as shown in Table 6.

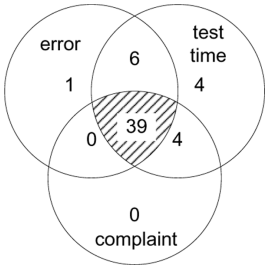


Fig. 4. Seen quality attributes of COSD (Q6a))

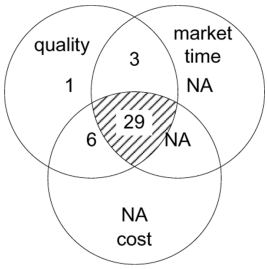


Fig. 5. Experienced advantages of COSD (Q6)

Table 6. Opinion on COSD as a better way to develop software from the component users (Q7)

Answer	Frequency	Percentage
Agree	55	94.83%
Disagree	3	5.17%

In the next subsection, analysis done on the feedback received from the non-component users is presented.

3.3 Non-component Users' Perception

From Fig. 3 it can be seen that 43 out of 101 respondents, which accounts for 42.57% of the total respondents, do not use components in their software development projects. From the non-component users, we determine the potential of COSD application by asking about their willingness to use components in their future software development project and their perception on COSD. When asked about their willingness to use components in future software development projects, if given the opportunity to do so, 90.70% of them will adopt COSD in the future if given the chance. Furthermore, 93.02% of the non-component users also agree that COSD is a better way to develop software. These results are shown in Table 7 and Table 8 respectively.

Table 7. Chances of using components in future software development projects (Q8)

Answer	Frequency	Percentage
Will use	39	90.70%
Will not use	4	9.30%

Table 8. Opinion on COSD as a better way to develop software from the non-component users (Q9)

Answer	Frequency	Percentage
Agree	40	93.02%
Disagree	3	6.98%

4 Discussion

The potential of COSD application can be determined from the responses given by the software developers in general, from the component users and from the non-component users as illustrated in Fig. 1. Therefore, the subsequent three paragraphs will discuss and conclude about the potential from the viewpoints of the three groups of respondents respectively.

From the current software development practice of all respondents, it can be seen that the inclination is towards software reuse. This is shown by the types of programming languages used, which are mainly object-oriented programming languages; the development methodology applied, which are mainly object-oriented; and the amount of code being reused where majority of them reuse at least 50% of the code from previous projects. COSD is designed to support reuse and this inclination is in line with COSD. Therefore, it can be concluded that from the viewpoint of the software developers in general, the potential of COSD application is high. The fact that majority of the software developers are already familiar with the term component further improves this potential.

Despite the many problems faced in using components, 94.83% of the existing component users still believe that COSD is a better way to develop software. This belief is most likely supported by the fact that half of them (50.00%) experienced all the advantages of COSD while using it. This means that the possibility of the component users to continue using this development approach is there. Therefore, from the viewpoint of the current component users, it can be concluded that the potential of COSD to be applied in their software development projects is also high.

From the side of the non-component users, the prospect is also very encouraging. 93.02% of the non-component users agree that COSD is a better way in developing software despite not applying them in their software development projects, with 90.70% of them will apply the approach if given the chance to do so. Therefore, we can also conclude that the potential of COSD to be applied by

the non-component users is also high. With these, we conclude that the potential of COSD application in Malaysia is high.

5 Related Work

In our review, at least three work that studied about the application of COSD were found. The first one covered the software companies in the Kingston city of Jamaica [2], the second one covered the software development projects in three European countries; Norway, Germany and Italy [12], and the third one was performed on the software development projects in China [13].

The first study, which was done based on the feedback from eight prominent software development companies in Kingston city of Jamaica covered:

- the level of components reuse in software development,
- the quality of software systems created with components reuse,
- the average number of software systems created per year and
- the cost associated with components reuse.

In particular, it concluded that all of the companies involved in the case study have experienced the benefits of COSD (i.e. improved quality, higher productivity and reduced development cost) and that the main success factor of COSD was attributed to the existence of good components repositories.

The second work performed an empirical study on off-the-shelf (OTS) component usage in industrial projects [12], which was performed on the software development companies in three European countries; Norway, Italy and Germany. OTS components in this study were categorised into commercial off-the-shelf (COTS) components and open source software (OSS) components. COTS components are owned by commercial vendors and their users usually do not have access to the source code of these components, whereas OSS components are provided by open source communities that offer full control of the source code. The study investigated a total of 71 projects that exclusively used COTS components and 39 projects that exclusively used OSS components. Three dimensions under investigation are:

- the users of OTS components,
- the reasons for deciding to use the OTS components and
- the outcomes of using the components.

The study concluded that the users of COTS and OSS components are companies with similar profiles where software houses and information technology consulting companies were the main users of these components. Meanwhile, the top three reasons for using the OTS components, both the COTS and OSS, are time-to-market, reliability and performance, with the key motivations being to save development time and effort and to get newest technology. The results of using OTS components were formulated in term of the risks experienced during projects development. Fifteen possible risks were investigated and the result

showed that some risks were common to both OTS and OSS component users and some others were different.

The third study investigated the software development using OSS components in Chinese Software Industry. This survey particularly focused on three issues in reusing OSS components. The issues are component selection, licensing terms and system maintenance. On component selection, they concluded that the selection of OSS components were made initially using existing web search engines, followed by local expertise for evaluation of the selected components. On licensing terms, the survey discovered that the OSS licensing terms were not seen as a barrier to the software companies in China when they reuse the components in their software development projects. Finally, with regard to system maintenance, they concluded that 84% of the maintenance work is dedicated to bug fixing or other code changes in the selected OSS components. Even though the developers admitted the need for active participation in the OSS community for this purpose, close participation with OSS community was still rare.

However, it is not easy to make any useful conclusion that relates these studies to ours, as each is focusing on different aspects and focus of component usage. In our situation, COSD is a relatively new research area and therefore, the focus is more on identifying the current practice of the software developers in the process of determining its potential.

6 Conclusion

In this paper, the outcomes of a study done to investigate the current state of COSD application amongst Malaysian software developers are presented. The study was accomplished through questionnaire distribution to the software developers who are working at various software development companies in Malaysia. The aim of the study is to determine the potential of COSD application in Malaysia. Analysis of the responses received shows that the opportunity of COSD application in software development amongst Malaysian software developers is high. Such a finding can further encourage the researchers in the field of COSD to intensify their research work and promotes its application, having known that the potential is high. It can also be used by the researchers as a basis to support the relevance of their related research work.

References

1. Prieto-Diaz, R.: The Disappearance of Software Reuse. In: 3rd International Conference on Software Reuse, p. 255. IEEE CS Press, New York (1994)
2. Pyne, R.S., McNamara, M., Bernard, M., Hines, D., Lawrence, G., Barton, D.: An Evaluation on the State of Component-based Software Engineering in Jamaica. In: IEEE Southeast Conference 2005, pp. 570–575. IEEE CS Press, New York (2005)
3. Lethbridge, T.C., Sim, S.E., Singer, J.: Studying Software Engineers: Data Collection Techniques for Software Field Studies. In: Empirical Software Engineering, vol. 10, pp. 311–341. Springer Science + Business Media, Inc, The Netherlands (2005)

4. Basili, V.R., Caldiera, G., Rombach, H.D.: Goal Question Metric Paradigm. In: *Encyclopaedia of Software Engineering*, pp. 528–532. John Wiley & Sons Inc., Chichester (1994)
5. Seta, F., Onishi, T., Kidokoro, T.: Study about Locational Tendency of IT Companies in City Centers and Suburbs - Case Study of Malaysia. In: *International Symposium on Urban Planning*, pp. 257–266 (2001)
6. MDec. List of MSC status companies (2007) (accessed April 13, 2007), <http://www.msomalaysia.my/topic/Company+Directory>
7. Raja Kassim, R.S., Kassim, E.S.: Knowledge Management Practices amongst MSC Status Companies in Malaysia: A survey. *International Journal of Knowledge, Culture and Change Management* 5(9), 63–70 (2000)
8. Schreiner, K.: Malaysia's silicon valley moves forward. *IEEE Software*, 126–130 (1999)
9. Lee, E.K.J.: Experience in ASEAN: Perspective from Software Consortium of Penang (SCoPe). In: *4th Engage European Union South East Asia ICT Research Collaboration Conference*, Penang, Malaysia (2007)
10. Pandi, A.R.: Web-Based Survey versus Conventional Survey: The Malaysian Experience in Conducting the Internet Subscriber Study. In: *International Conference on Improving Surveys*, Copenhagen, Denmark (2002)
11. Aris, H., Salim, S.S.: Issues on the Application of Component-Oriented Software Development: Formulation of Research Areas. *Information Technology Journal* 7(8), 1149–1155 (2008)
12. Li, J., Torchiano, M., Conradi, R., Slyngstad, O.P.N., Bunse, C.: A State-of-the-Practice Survey of off-the-shelf Component-based Development Processes. In: Morisio, M. (ed.) *ICSR 2006. LNCS*, vol. 4039, pp. 16–28. Springer, Heidelberg (2006)
13. Chen, W., Li, J., Ma, J., Conradi, R., Ji, J., Liu, C.: A survey of Software Development with Open Source Components in Chinese Software Industry. In: Wang, Q., Pfahl, D., Raffo, D.M. (eds.) *ICSP 2007. LNCS*, vol. 4470, pp. 208–220. Springer, Heidelberg (2007)
14. Conradi, R., Li, J., Slyngstad, O.P.N., Kampenes, V.B., Bunse, C., Morisio, M., Torchiano, M.: Reflections on conducting an international survey of software engineering. In: *Proceedings of the International Symposium on Empirical Software Engineering*, pp. 214–223. IEEE CS Press, New York (2005)

On Using Semantic Transformation Algorithms for XML Safe Update

Dung Xuan Thi Le and Eric Pardede

Department of Computer Science and Computer Engineering
La Trobe University, Melbourne VIC 3083, Australia
{dx113@students., E.Pardede@}latrobe.edu.au

Abstract. XML update support in data repositories is gaining a new level of importance since the use of XML as data format has been widely accepted in many information system applications. In addition to the capabilities of update, research on how the query update is performed is also needed. Safe update maintains the integrity of the updated XML documents, which can be costly. In this paper, we show how to improve the performance of the safe updates using a series of semantic transformation algorithms. The algorithms will pre-process the schema to preserve the documents' semantics/constraints. The information gathered is used to transform the update requests into semantic updates that can outperform the primitive safe updates.

Keywords: XML Updates, XML Schema, XQuery, XPath.

1 Introduction

XML Update once was perceived as an unnecessary operation in XML data management. XML documents were considered static and even if an update was necessary, the users simply replaced a whole document with a new one. However, with the increased usage of XML as a data format, there is a new attitude amongst XML communities towards XML updates. As with any traditional database, XML documents should be updateable and all issues associated with update operations have become emerging research topics.

Many information system applications also use XML as the underlying data representation for their business. According to a recent study [3], 60% of information is now structured in XML-based representation. In addition, the use of XML encoding for standard data specification standards, such as BPEL (Business Process Execution Language), HL7 (Health Language 7) and AIXM (Aeronautical Inter Exchange Model) etc. has flourished.

The volume of research on XML Updates has increased over the last few years, ranging from proposals for the updating of languages to studies of the implications of XML updates. Since XML documents can be stored in different repositories, the research on XML updates also varies based on the repository types.

A common understanding between different research on XML updates is that XML safe update is an expensive operation. However, since XML update has now become

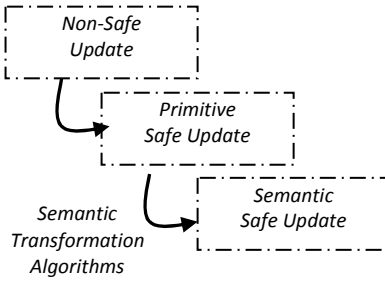


Fig. 1. Problem Definition

a core requirement and not merely an option, XML communities have to perform XML safe update during their data management. The question is how to find the cheapest way of performing the safe update.

In this paper, we aim to improve XML safe update performance by using semantic transformation algorithms (see Fig.1). The updates take form in basic operations such as deletion, replacement and renaming of XML element(s)/ attributes(s). The updates will be the safe update, which considers the conceptual semantics/constraints of the documents, yet performance should improve from the primitive safe update operations [8]. For the repository, we use an XML-Enabled Database.

Roadmap. Following the introduction, in section 2, we provide an overview of XML update and related works. In section 3, we discuss the framework of our work in comparison to existing work on XML update. Our update algorithms are explained in section 4 and we discuss the implementation setting as well as the case study in section 5. We perform some analysis in section 6 and conclude this paper in section 7.

2 XML Updates

2.1 Overview

In the early days of using XML as data format, there was no capability for updating the data. If one wanted to change an XML data, one needed to remove the current version and upload a newer version of the data into the repository. This is, of course, not acceptable nowadays, as XML data has become more dynamic, in terms of the frequency of changes.

[13] has proposed a data manipulation language for the XML database. In this paper, the authors proposed extensions to XQuery to support XML updates. The updates only consider the instances and therefore, no schema checking or constraint preservation is enforced. We refer to these as *non-safe updates*, as opposed to *safe updates*.

Safe updates, on the other hand, consider schema checking or validation before performing the operations. In this paper, we will address three basic updates as described below. For a running example, we use the XML schema and the instance in Figure 2. The schema is shown to describe the constraints, while the instance is shown to demonstrate the queries in this paper.

- Insertion involves the addition of node(s) or attribute(s) into the XML data in the database. For example (see Fig.2), *insert a second address of a particular employee*.
- Deletion involves the removal of node(s), attribute(s) or their contents from existing data. For example (see Fig.2), *removal of all contract employees who work in Melbourne*.
- Replacement involves the change of node(s) and attribute(s). The change can be of the name or the content. For example (see Fig.2), *change the supervisor of a particular employee*.

In the next section, we describe some existing work on XML updates to show how our work fits in and contributes to the research area of XML update.

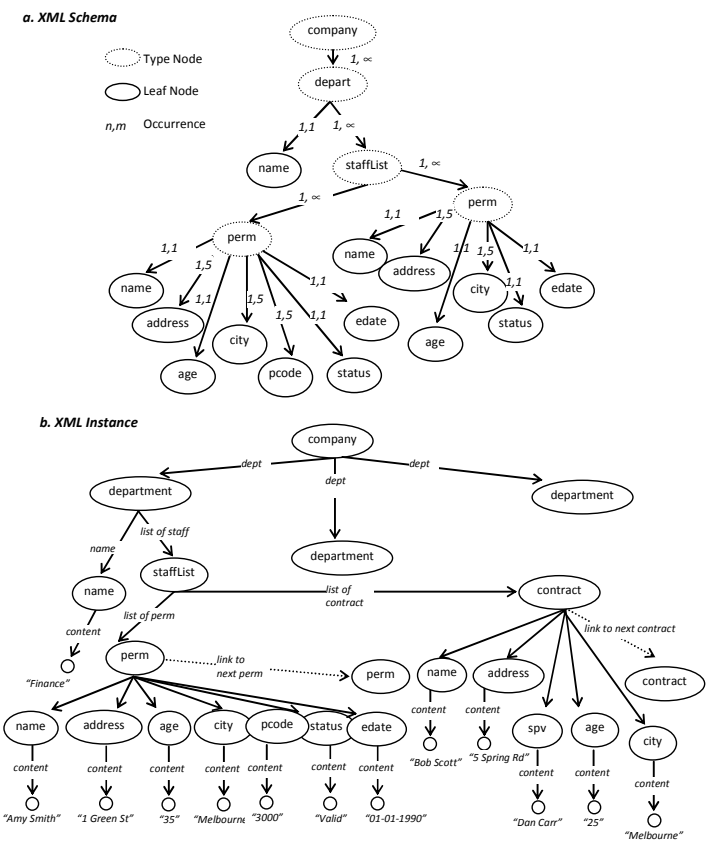


Fig. 2. Sample XML: Schema and Instance

2.2 Related Works

XML update mechanisms differ depending on the storage of the XML data. For Native XML Database (NXD), many products use proprietary languages, such as Lore

[4], that allow updating within the server. Other products use special languages such as XUpdate [7], which are designed to be used independently for any kind of implementation. Another strategy that is followed by most NXD is to retrieve the XML document and then update the XML using XML API [6, 10].

Another option for XML update in NXD is to embed the update processes into XML language and enable the query processor to read the update queries. This is the latest development and interest in this is growing. The first work in this area [13] embedded simple update operations into XQuery, thus it can be used for any NXD that supports this language. Subsequent work has extended this proposal; however, a basic issue remains unresolved, that being, we do not know how the update operations may affect the semantic correctness of the updated XML documents.

Recently, W3C released a working draft for update facilities in XQuery [14] which is a first step towards fully integrating the update operations in available XML databases.

When we use an XML-Enabled Database (XED) for XML storage, we have the benefits of the full database capability, including full support for update processes. Work has been conducted in some DBMS to translate XML query languages (and updates too, for that matter) into DBMS query languages such as SQL, the idea being to cover the expressive power of the XML query languages into a more widely used SQL language [1, 2].

The release of the SQL/XML standard [5] has provided a uniform language to manage the XML data in XED. Many XED products now implement update facilities that comply with this latest standard [12]. Unfortunately, there is no clear discussion on how to carry out safe manipulation by checking the XML constraints.

In this work, we will propose algorithms to support safe manipulation, a feature which has been missing from existing work. The implementation of our algorithms will be based on the SQL/XML data manipulation language. In the next section, we show the syntaxes for the updates using this standard.

2.3 XML Updates Using SQL/XML

The release of the SQL/XML standard has provided developers and users of XED with a relational platform, a common language for XML data management. Due to our page limitation, we only show a sample of standard statements for data manipulation provided in [5].

For deletion:

```
DELETE [WITH <XML lexically scoped options>]
FROM <target table>
[WHERE <search condition>]
```

For insertion:

```
INSERT [WITH <XML lexically scoped options>]
INTO <insertion target>
<insert columns and source>
```

For update:

```
UPDATE [WITH <XML lexically scoped options>]
<target table>
SET <set clause list>
[WHERE <search condition>]
```

Understandably, the implementation of the standard can differ based on the environment. In the product we use for implementing our algorithms, the `<XML lexically scoped options>` is the conditional statement written in XPath. The same is applied to the `<set clause list>` and `<search condition>` components.

Every time a manipulation request is made using one of the statements above, the database will proceed without checking the impact of the operations on the integrity of the data. This is what we refer to as a *non-safe update*. In contrast, we want to support safe updates by providing the mechanisms to check the constraints before the manipulation takes place.

3 Safe Updates Framework

3.1 Primitive Safe Update Framework

Authors of previous work [11] have demonstrated how primitive safe updates can support constraint preservation of updated XML documents. However, these incur high costs compared to primitive updates [9].

Figure 3a shows the framework of primitive safe updates. Every time there is an update request, a trigger¹ is employed to check the effect of the update on the constraints stated in the XML Schema. Based on the results of the check, the system either confirms the update or orders the cancellation of the update.

Using this approach, high costs are incurred in accessing XML Schema and processing the update validity for every single XML Update. Therefore, a more efficient way to perform safe update is necessary. One solution is to incorporate the semantic transformation mechanism [8].

3.2 Semantic Safe Update Framework

The framework of semantic safe update is designed and shown in Figure 3b. On the start-up, the pre-processing schema is initiated so that all constraints/semantics defined in the given schemas are processed. This is shown as A in Figure 3b.

The semantics are housed in the transformer component. The pre-processing schema component is no longer needed here, hence it is terminated. It only restarts if the schema is changed/modified.

The transformer is a real-time system component. Its functionality is to accept the XML input predicate (in our case, as an XPath) and verify the constraints and transform it where possible. If the path predicate is valid, the transformer sends its semantic path predicate to the XED to perform the update. Otherwise, the conflict is detected and an informative message is returned to the user by the transformer. This is shown as B in Figure 3b.

As we can see from the framework, our methodology offers very minimal interaction with the database. Also, the methodology provides the advanced ability to determine whether or not the query needs to access the database during the transformation phase.

¹ A built-in trigger can be bundled with RDBMS package.

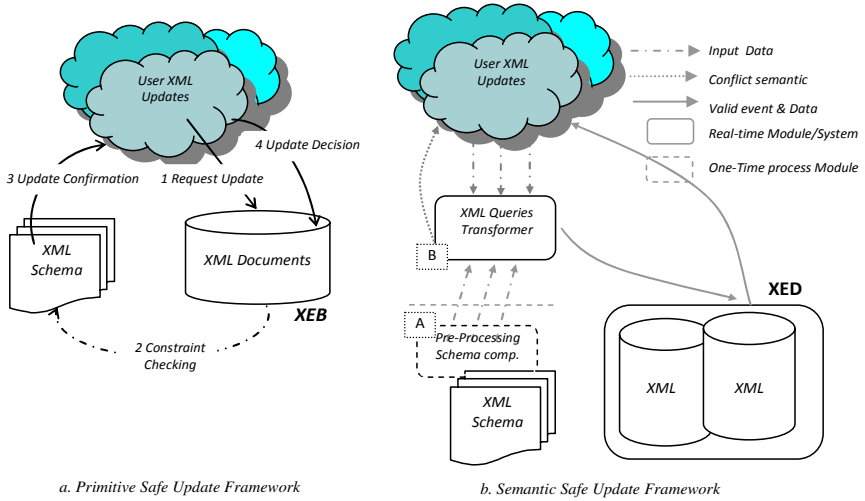


Fig. 3. Safe Update Frameworks

In the next section, we detail the proposed algorithms and how they rewrite the primitive safe update query into the semantic safe update query.

4 Proposed Algorithms

As shown in Figure 3, two main tasks are conducted in semantic safe update: task A is the pre-processing of schema and task B is the transformation of the XML queries.

Algorithm 1 is used to perform task A. This algorithm accepts two input parameters including a valid XML schema and the name of the schema root. The schema is processed and two data structure lists S and C are built respectively (Line 1-5 to 1-6). S stores a sequence of elements in the form of $[parent][child]$ extracted from the schema. C stores a series of leaf-nodes and constraints followed by a series of restricted values in the form of $[leaf\ node\ name][constraint\ name][V]$, where V is a set of restricted values assigned to each constraint that is associated with a leaf-node.

For example, a leaf node “pcode” is restricted to an acceptance range, “inclusive” constraint, of between 2000 and 4000. For this particular node, we refer to the various constraints as *cardinality*, *inclusive*, *exclusive*, *enumeration*, *length* and *pattern*. All of these constraints will be processed by algorithm 1 and it will return S, C lists.

Upon completion of schema pre-processing, the Semantic Transformation Algorithm (*Algorithm 2*) takes control. The goal of this algorithm is to transform the XPath into a semantic XPath.

Algorithm 2 is started with a derivation of a list of unique paths² (Line 2-9) from the sequence element list S . As we deal with update query type, the XPath is expected

² A unique path is a full path that traverses from the root of the XML schema tree to a selected element. The path elements are separated by operator “/”.

Algorithm 1: Pre-processing Schema

```

1-1: Input
1-2:    $\mathcal{T}$  = XSD schema
1-3:    $\mathcal{R}$  = root name of the schema
1-4: Output
1-5:    $\mathcal{S}$  = List of sequence elements defined in XSD Schema.
1-6:    $\mathcal{C}$  = semantic knowledge of elements obtained from  $\mathcal{T}$ 
1-7: Begin
1-8:   push("root",  $\mathcal{R}$ ) into  $\mathcal{S}$ 
1-9:   Let  $eType$  = List of types. If an element is a type then not leaf node
1-10:  push ( $\mathcal{R}$ ) into  $eType$ .  $\mathcal{R}$  must be a type as it has at least one child.
1-11:  WHILE  $eType$  not empty read next line  $\ell$ 
1-12:    IF pop ( $eType[0]$ )  $\in \ell$  THEN  $eType[0]$  is 1st element of  $eType$ 
1-13:    REPEAT read next line to get  $f_i$  child of  $eType[0]$ 
1-14:    push( $e_i, f_i$ ) into  $\mathcal{S}$ 
1-15:    IF  $f_i$  has valid semantic
1-16:    push( $f_i, n, \mathcal{V}$ ) into  $\mathcal{C}$  where  $n$  is the name of constraint;  $\mathcal{V}$  is a series of constraint values
1-17:    IF  $f_i$  is also a type push ( $f_i$ ) into  $eType$ 
1-18:    UNTIL encounter end of  $eType[0]$ 
1-19:  Return  $\mathcal{S}, \mathcal{C}$ 

```

with some restrictions in order to process the update. From these restrictions, we identify the elements, their values or their paths (Line 2-10 to 2-21). We refer to these types of paths as *sub-paths*.

The identified restricted elements are then verified against the list of elements in the left-node element list $\mathcal{C}$ (Line 2-11 to 2-18). In the case of the restricted elements being found in list $\mathcal{C}$, the elements' restricted values will then be verified (Line 2-12). For example, the 'age' element is a restricted element and has a constraint value range between 16 and 52. If the restricted value of 'age' in the predicate is greater than 15 or less than 53, the predicate can be removed from Q (Line 2-12). On the other hand, if it is less than 15, then the transformed XPath is set to 'conflict' and no further transformation is needed. An alert is returned to notify the user. This also applies to the element in the predicate that does not exist in the leaf-node list $\mathcal{C}$ or the $\mathcal{S}$ list.

In the case where the predicate contains only a restricted path, the algorithm also performs the transformation. If the restricted path is not expressed as a full path, which means it contains only a sequence of elements and operator '/', then it has to be transformed so that it can be a full sub-path of one of the unique paths (Line 2-20). We use *fn_semantic_path_transformation* function for transformation (Line 2-20) to complete the path transformation task.

The *fn_semantic_path_transformation* is comprised of two operations, namely *semantic expansion* and *semantic contraction*, which were firstly introduced in [7]. The former is a transformation of a given XPath to a unique path, while the latter is a transformation of a given XPath to another XPath, which is preceded by operator "//", hence it is contracted from the recursive type in the XPath.

In this paper, the *fn_semantic_path_transformation* has been selectively adopted to transform any path, including the restricted path, to a semantic restricted sub-path or a semantic path.

Algorithm 2: Semantic Transformation

```

2-1: Input
2-2:    $\mathcal{S}$  = List of sequence elements defined in XSD Schema.
2-3:    $\mathcal{C}$  = Semantic knowledge of elements obtained from
2-4:    $Q$  = XPath
2-5: Output
2-6:    $\wp$  = Semantic Xpath or Error message
2-7: Begin
2-8:   Repeat
2-9:     Let  $\mathcal{U}$  be a list of unique path derived from  $\mathcal{S}$ ,  $r$  be the inner focus of predicate  $[\ ]$ ,
        $Q$  be xpath input by user,  $\mathcal{P}$  is set to NULL
2-10:    FOR each  $r$  in ( $Q \neq \text{NULL}$ ) DO
2-11:      Let  $\gamma$  be restricted element in  $r$ ,  $\phi$  be restricted values in  $r$ ,  $\tau$  be fragment of sub-path in  $r$ 
2-12:      IF  $\gamma$  found in  $\mathcal{C}$  list and all values  $\phi \in \gamma$  existed  $r$  are in the domain range
        of/equivalent to all values  $\phi \in \gamma$  existed in  $\mathcal{C}$  THEN
2-13:        remove  $r$  from  $Q$ 
2-14:      ELSE IF  $\gamma$  found in  $\mathcal{C}$  list and only some value  $\phi \in \gamma$  in the domain range
        of/equivalent to some values  $\phi \in \gamma$  existed in  $\mathcal{C}$  THEN
2-15:        retain  $r$  in  $Q$ 
2-16:      ELSE IF  $\gamma$  not found in  $\mathcal{C}$  list next item or  $\gamma$  found and  $\phi \in \gamma$  not in the domain range of  $\phi \in \gamma$  in  $r$  THEN
2-17:        set  $\wp$  = 'conflict'
2-18:      EXIT FOR
2-19:      IF found  $\tau$  as a sub-path in  $\mathcal{U}$  list item AND  $\tau$  contains only operator ' $/$ ' THEN remove  $\tau$  from  $Q$ 
2-20:      IF found  $\tau$  as a sub-path in  $\mathcal{U}$  list item AND
         $\tau$  contains operators other than ' $/$ '  $\tau' = \text{call } fn\_semantic\_path\_transform(\tau)$ 
2-21:      ELSE IF not found  $\tau$  as a sub-path in  $\mathcal{U}$  list item or  $\tau'$  is empty THEN set  $\wp$  = 'conflict'
2-22:       $\wp = Q$  where  $Q$  has been transformed
2-23:      IF  $\wp$  is an UPDATE query THEN
2-24:        WHILE not end of  $\mathcal{U}$ 
2-25:          Match  $\phi$  to  $\mathcal{C}$  list next item where  $\phi$  is the update element
2-26:          IF  $\phi$  found in  $\mathcal{C}$  list next item and value of  $\phi$  matched value of found next item THEN
2-27:             $\wp = \text{call } fn\_semantic\_path\_transform(\wp)$ 
2-28:            EXIT
2-29:          ELSE  $\phi$  not found in  $\mathcal{C}$  list next item or  $\phi$  found in  $\mathcal{C}$  list next item and value of  $\phi$  not matched THEN
2-30:             $\wp = \text{'conflict'}$ 
2-31:      ELSE IF  $\wp$  is a DELETE query THEN
2-32:         $\wp = \text{call } fn\_semantic\_path\_transform(\wp)$ 
2-33:      IF  $\wp$  is not NULL AND  $\wp$  is not 'conflict' THEN send  $\wp$  to access the database to complete its task
2-34:       $\wp = \text{'Update/DELETE Done'}$ 
2-35:    UNTIL Input is 'Esc' Key '

```

Continuing with the algorithm, Line 2-19 allows a predicate if the restricted path is a sub-path of one of the unique paths in the list derived from the $\mathcal{S}$. The restricted path will also cause semantic conflict where it does not match the schema structure and where it is not a sub-path of any unique path (Line 2-21).

Finally, Line 2-23 to Line 2-32 perform the transformation of the actual update queries where the values used for update are ensured to be within the restricted values in the schema. This operation ensures that the path is at its best efficiency and performance before it is sent to the database.

5 Implementation

We describe the implementation into two stacks: (1) the *hardware stack* includes a machine that has a configuration of AMD Athlon 64 3200+, with 2300 MHz and 2.0

GB of RAM; and (2) the *software stack* includes a Windows XP Professional OS and Java VM 1.5. We select a leading commercial XED and use their provided database connection driver to connect to our algorithm modules.

We use five synthetic datasets (compliant with the schema shown in Fig.2) of varying sizes: 5, 10, 15, 20 and 25 megabytes. We run various update queries for each data set. Due to space limitations, we show only three of the queries. For each query, we show the original query and the result of query rewriting after it goes through the semantic transformation algorithms.

Query 1: Updating text element with a conditional predicate - Valid operation

Original Query:

```
UPDATE companyxml5
SET OBJECT_VALUE = UPDTEXML(OBJECT_VALUE, '//perm/status/text()',
'Invalid')
WHERE existsNode(OBJECT_VALUE, '//perm[ages<35]')=1;
```

Transformed Query:

```
UPDATE companyxml5
SET OBJECT_VALUE = UPDTEXML (OBJECT_VALUE,
'/company/department/staffList/perm/status
/text()', 'Invalid')
WHERE existsNode(OBJECT_VALUE,
'/company/department/staffList/perm[ages<35]')=1
```

Query 2: Deleting node element with a conditional predicate - Valid Operation

Original Query:

```
UPDATE companyxml5 set object_value = DE-
LETXML(OBJECT_VALUE, '/company/department/staffList/*[edate <"01-
01-1980"]');
```

Transformed Query:

```
UPDATE companyxml5 set object_value = DE-
LETXML(OBJECT_VALUE, '/company/department/staffList/perm[edate
<"01-01-1980"]')
```

Query 3: Updating text element with a conditional predicate - Invalid Operation

Original Query:

```
UPDATE companyxml5 SET OBJECT_VALUE = UPDTEXML (OBJECT_VALUE,
'//contract/status/text()', 'Invalid')
WHERE existsNode(OBJECT_VALUE, '//perm[ages<15]')=1
```

Transformed Query:

No query - conflict ages constraint 16<ages<55

As detailed in the previous section, the schema has been pre-processed prior to the update queries. From the series of queries above, the first and second queries showed valid updates. The third query is an invalid update, which results in no semantic query being produced and hence there is no need to access the database.

6 Analysis

Previous work [9] has shown how non-safe update significantly outperforms primitive safe update. This result is not surprising since in non-safe updates, there is no constraint checking involved during the operations. The negative side is that this can violate the original constraints of the document.

In this paper, we compare the performance of primitive safe update and semantic safe update (see Fig.4).

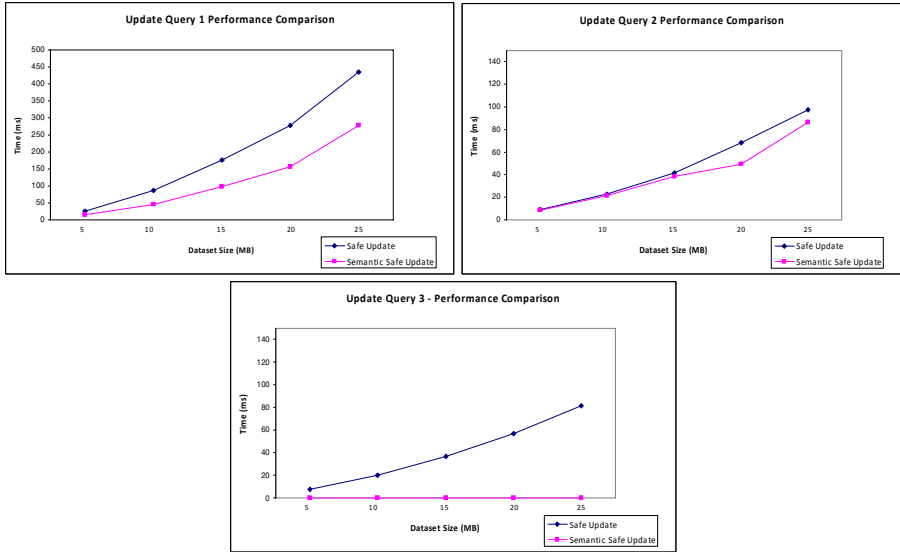


Fig. 4. Primitive Safe & Semantic Safe Update Performance Comparison

In query 1, we demonstrate the descendant-or-self ‘//’ operator using the structural and simple values constraint of an element in the predicates. In the first instance, we validate the path predicate of the query. The validation aims to check whether the element in the predicate confirms the value stated in the schema. It also checks whether the structural restricted path and the path being updated are valid. As the outcome, the paths are transformed to full paths and the semantic query performance is significantly improved by almost 60% compared to the performance of the original XPath.

In query 2, we use the union type “*” in the query. The query predicate does not involve any path restriction. Therefore, the return data is only restricted by simple elements with values. Our transformation is able to determine and map “*” to “/perm” as the schema constraint enforces the permanent employment date as compulsory for every permanent staff member (*edate* minOccur=1). The performance result shows a significant improvement when the data size growth is more than 20 MB. However, the effects of “*” still has to be further investigated since the curve of the graph shows that the performance is not always determined by the size of the data.

Finally, query 3 demonstrates the ability of our algorithm to detect the conflict constraint values at the transformation stage. As a result, access to the database is avoided and the update is not validated. By way of comparison, in the primitive safe update method, to determine the conflicts in constraint values, the query must be sent to the database for schema validation.

It is important to highlight that for this experiment we do not include the schema pre-processing time for two reasons: (i) the operation is only done once on the start-up and the processed constraints are re-used by all the queries unless the schema is changed/modified; (ii) the processing time is based on the complexity of the schema. To pre-process the schema, memory is one of the main considerations. Luckily, nowadays, the cost of memory is not a problem and it can be added anytime.

This analysis has shown that our semantic safe update can contribute to better performance in XML safe update, without sacrificing the constraints integrity of the updated documents.

7 Conclusion

With a large proportion of data in business applications now represented as XML, more efficient methods of managing XML data in various XML storages are urgently needed. A facility for updating XML data had been lacking in XML data management research. Despite the fact that some recent work on proposing safe updates has been conducted, these are usually very costly in comparison to non-safe updates.

To solve this problem, in this paper, we have proposed semantic transformation algorithms for semantic safe update in XML database. As a result, every valid update implemented using the algorithms has achieved an improvement of 20-100% in comparison with primitive safe updates.

In our future work, we will continue to extend our algorithms to handle more complex semantics that have not been addressed in this paper, such as the conjunctive and the disjunctive restrictions in predicates. In addition, checking of occurrence constraints on instances – instead of in schema information only – may be investigated.

References

1. DeHaan, D., Toman, D., Consens, M.P., Özsu, M.T.: A Comprehensive XQuery to SQL Translation using Dynamic Interval Encoding. In: SIGMOD, pp. 623–634 (2003)
2. Du, F., Amer-Yahia, S., Freire, J., ShreX: Managing XML Documents in Relational Databases. In: VLDB, pp. 1297–1300 (2004)
3. Gartner, Inc., Gartner's 2006 Emerging Technologies Hype Cycle Highlights Key Technology Themes (August 2006), <http://www.gartner.com>
4. Goldman, R., McHugh, J., Widom, J.: From Semistructured Data to XML: Migrating the Lore Data Model and Query Language. In: WebDB, pp. 25–30 (1999)
5. ISO/IEC. Information Technology – Database Languages – SQL – Part 14: XML-Related Specifications (SQL/XML). ISO/IEC 9075-14 (2006)

6. Jagadish, H.V., Al-Khalifa, S., Chapman, A., Lakshmanan, L.V.S., Niernan, A., Paprizos, S., Patel, J.M., Srivastava, D., Wiwattana, N., Wu, Y., Yu, C.: TIMBER: A native XML database. *VLDB Journal* 11(4), 279–291 (2002)
7. Laux, A., Martin, L.: XUpdate Working Draft (September 14, 2000) (Accessed July 20, 2008), <http://xmldb-org.sourceforge.net/xupdate/>
8. Le, D.X.T., Bressan, S., Rahayu, J.W., Taniar, D.: Semantic XPath Query Transformation: Opportunities and Performance. In: Kotagiri, R., Radha Krishna, P., Mohania, M., Nantajeewarawat, E. (eds.) *DASFAA 2007*. LNCS, vol. 4443, pp. 994–1000. Springer, Heidelberg (2007)
9. Le, D.X.T., Pardede, E.: Towards Performance Efficiency in Safe XML Update. In: Benattallah, B., Casati, F., Georgakopoulos, D., Bartolini, C., Sadiq, W., Godart, C. (eds.) *WISE 2007*. LNCS, vol. 4831, pp. 563–572. Springer, Heidelberg (2007)
10. Pardede, E., Rahayu, J.W., Taniar, D.: XML Update Management in XML-Enabled Relational Database. *Journal of Computer and System Sciences* 74(2), 170–195 (2008)
11. Oracle. A Comparison of Oracle Berkeley DB and RDBMS Sleepy Cat: Berkeley DB XML (Accessed July 20, 2008), <http://www.oracle.com/database/docs/Berkeley-DB-v-Relational.pdf>
12. Scardina, M., Chang, B., Wang, J.: *Oracle Database 10g XML & SQL: Design, Build & Manage XML Applications in Java, C, C++, & PL/SQL*. McGraw Hill, Osborne (2004)
13. Tatarinov, I., Ives, Z.G., Halevy, A.Y., Weld, D.S.: Updating XML. In: *SIGMOD*, pp. 413–424 (2001)
14. W3C. XQuery Update Facility 1.0. W3C Candidate Recommendation (2008), <http://www.w3.org/TR/xquery-update-10/>

Storing and Querying Graph Data Using Efficient Relational Processing Techniques

Sherif Sakr

NICTA and University of New South Wales
Sydney, Australia
`Sherif.Sakr@nicta.com.au`

Abstract. Graphs have become increasingly used for modelling complicated data such as: chemical compounds, protein interactions and social networks. Retrieving related graphs containing a query graph from a large graph database is a fundamental performance issue in any graph-based application. Relational database management systems (RDBMSs) have repeatedly shown their success and efficiency in hosting types of data which have formerly not been anticipated to live inside relational databases such as: complex objects and XML data. The big advantages of relational database systems are its well-known maturity and its high scalability to handle vast amounts of data very efficiently. In this paper, we investigate the efficiency of different proposed schemes for storing and querying various kind of graphs using the relational infrastructure. Moreover, we investigate how existing relational query optimization techniques could be effectively utilized to improve the processing times of relational-based processing of graph queries. Finally, we have qualitatively evaluated our proposed approaches using an extensive set of experiments.

1 Introduction

Graphs are among the most complicated and general form of data structures. Recently, they have been widely used to model many complex structured and schemaless data such as XML documents [16], protein networks [5], social networks [4] and chemical compounds [14]. Retrieving related graphs containing a query graph from a large graph database is a key performance issue in all of these graph-based applications. Therefore, efficient query engines that allow users to effectively store and query graph data is crucial to exploiting the full power of graph data. For persistent storage of graph data, naturally using Relational database management systems (RDBMSs) comes to mind. RDBMSs have repeatedly shown that they are very efficient, scalable and successful in hosting types of data which have formerly not been anticipated to live inside relational databases such complex objects [8], spatio-temporal data [9] and XML data [6,13]. In addition, RDBMSs have shown its ability to handle vast amounts of data very efficiently using its powerful indexing mechanisms. In principle,

RDBMSs derive much of their performance from sophisticated optimizer components which makes use of physical properties that are specific to the relational model such as: sortedness, proper join ordering and powerful indexing mechanisms. In this paper we focus on employing the powerful features of the relational infrastructure to implement efficient mechanisms for processing graph queries. We present cost-effective relational-based mechanisms for processing graph queries on various kind of graphs. In our approach, the graph data set is firstly encoded using *fixed* relational storage schemes then the graph queries are translated into SQL queries over the defined storage schemes. An obvious problem in the relational-based evaluation approach of graph queries is the huge cost which may result from the large number of join operations which are required to be performed between the encoding relations. Therefore, we employ an effective and efficient pruning strategy to *filter* out as many as possible of the false positives graphs that are guaranteed to be not existing in the final results first before passing the candidate result set to the verification process. Specifically, we keep statistical information about the existing nodes and edges in the graph database in the form of simple Markov Tables [2]. This statistical information is used to influence the decision of the relational query optimizers by selectivity annotations of the translated query predicates to make the right decisions regarding selecting the most efficient join order and the cheapest execution plan. Consequently, it enables the query optimizers to get rid of the *non-required* graphs very early out of the intermediate results. Moreover, we carefully exploit the fact that the number of distinct vertices and edges labels are usually far less than the number of vertices and edges respectively. Therefore, we try to achieve the maximum performance improvement for our relation execution plans by utilizing the existing powerful *partitioned B-trees* indexing mechanism of the relational databases [11] to reduce the access costs of the secondary storage to the minimum [12].

1.1 Related Work

Recently, graph database has attracted a lot of attentions from the database community. In [10], Shasha et al. have presented *GraphGrep* as a path-based approach for processing graph queries. It enumerates all paths through each graph in a database until a maximum length L and records the number of occurrences of each path. An index table is then constructed where each row stands for a path, each column stands for a graph and each entry is the number of occurrences of the path in the graph. The main problem of this approach is that many false positive graphs could be returned in the filtering phase. In addition, enumerating the graphs into a set of paths may cause losing some of their structural features. Some researchers have focused on indexing and querying graph data using data mining techniques such as: *GIndex* [15], *TreePi* [3] and *Tree+Δ* [7]. In these approaches data mining methods are firstly applied to extract the frequent subgraphs (*features*) and identify the graphs in the database which contain those subgraphs. Clearly, the effectiveness of these approaches depends on the quality of the selected features. In addition, the index construction time of

these approach requires an additional high space cost and time overhead for enumerating all the graph fragments and performing the graph mining techniques. Moreover, all of these approaches deal with relatively small graph databases where they assume either implicitly or explicitly that the graph databases can completely or the major part of them fit into the main memory. None of them have presented a persistent storage mechanism for graph databases. In [17] Jiang et al. proposed another graph indexing scheme called *GString*. This approach is mainly focusing on decomposing chemical compounds into basic structures that have semantic meaning in the context of organic chemistry. The graph search problem is converted into a string matching problem and specific string indices is built to support the efficient string matching process. We believe that converting graph queries into sting matching problem could be an inefficient approach specially if the size of the graph database or the graph query is large. Additionally, it is not trivial to extend GString approach to support processing of graph queries in other domain of applications.

1.2 Organization of the Paper

The remainder of the paper is organized as follows: We discuss some background knowledge in Section 2. Sections 3 and 4 describe our relational-based mechanism for storing and querying *directed* and *undirected* label graphs respectively. Our techniques for optimizing the relational evaluation of graph queries are described in Section 5. We evaluate the performance of our proposed approaches by conducting an extensive set of experiments which are described in Section 6. Finally, we conclude the paper in Section 7.

2 Preliminaries

2.1 Labelled Graphs

Graph data structure is used to describe *relationships* among a set of *entities*. In labelled graphs, vertices and edges represent the entities and the relationships between them respectively. The attributes associated with these entities and relationships are called labels. A graph database D is a collection of member graphs $D = \{g_1, g_2, \dots, g_n\}$ where each member graph g_i is denoted as (V, E, L_v, L_e) where V is the set of vertices; $E \subseteq V \times V$ is the set of edges joining two distinct vertices; L_v is the set of vertex labels and L_e is the set of edge labels. In principal, labelled graphs can be classified according to the direction of their edges into two main classes: 1) *Directed-labelled graphs* such as XML and RDF. 2) *Undirected-labelled graphs* such as social networks and chemical compounds.

2.2 Graph Containment Queries

One of the classical graph query problems is to find all *supergraphs* of the query graph from a graph database. In principal, the graph containment query can

be simply described as follows: given a graph database $D = \{g_1, g_2, \dots, g_n\}$ and a graph query q , it returns the query answer set $A = \{g_i | q \subseteq g_i, g_i \in D\}$. A graph q is described as a sub-graph of another graph database member g_i if the set of vertices and edges of q form subset of the vertices and edges of g_i . To be more formal, let us assume that we have two graphs $g_1(V_1, E_1, L_{v1}, L_{e1})$ and $g_2(V_2, E_2, L_{v2}, L_{e2})$. g_1 is defined as sub-graph of g_2 , if and only if:

- 1) For every distinct vertex $x \in V_1$ with a label $vl \in L_{v1}$, there is a distinct vertex $y \in V_2$ with a label $vl \in L_{v2}$.
- 2) For every distinct edge $ab \in E_1$ with a label $el \in L_{e1}$, there is a distinct edge $ab \in E_2$ with a label $el \in L_{e2}$.

3 Relational Processing of Directed Labelled Graphs

3.1 Vertex-Edge Mapping Scheme

Our first proposed relational scheme for storing *directed labelled graphs* is the *Vertex-Edge* mapping scheme. In this mapping scheme, each graph database member g_i is assigned a unique identity *graphID*. Each vertex is assigned a sequence number (*vertexID*) inside its graph. Each vertex is represented by one tuple in a single table (*Vertices table*) which stores all vertices of the graph database. Each vertex is identified by the *graphID* for which the vertex belongs to and the *vertex ID*. Additionally, each vertex has an additional attribute to store the vertex label. Similarly, all edges of the graph database are stored in a single table (*Edges table*) where each edge is represented by a single tuple in this table. Each edge tuple describes the graph database member which the edge belongs to, the *ID* of the *source vertex* of the edge, the *ID* of the *destination vertex* of the edge and the edge label. The relational scheme of the *Vertex-Edge* mapping scheme is described as follows:

- *Vertices*(*graphID*, *vertexID*, *vertexLabel*)
- *Edges*(*graphID*, *sVertex*, *dVertex*, *edgeLabel*)

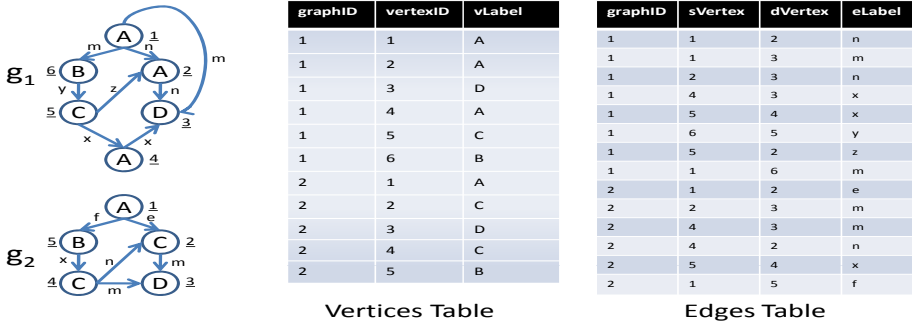


Fig. 1. Vertex-Edge relational mapping scheme of directed label graphs

Figure 1 illustrates an example of the Vertex-Edge mapping scheme of graph databases. In this figure, each underlined number in the representation of the directed labelled graph database members represents a *vertex ID*. Using these mapping scheme, we employ the following SQL-based *filtering-and-verification* mechanism to speed up the search efficiency of the graph queries.

Filtering phase: In this phase we specify the set of graph database members that may contain the set of vertices and edges which are describing the subgraph query. Therefore, the filtering process of a graph query q consists of a set of vertices QV with size equal m and a set of edges QE equal n can be achieved using the following SQL translation template:

```

1 SELECT DISTINCT  $V_1.graphID, V_i.vertexID$ 
2 FROM Vertices as  $V_1, \dots$ , Vertices as  $V_m$ , Edges as  $E_1, \dots$ , Edges as  $E_n$ 
3 WHERE  $\forall_{i=2}^m (V_1.graphID = V_i.graphID)$ 
4 AND  $\forall_{j=1}^n (V_i.graphID = E_j.graphID)$ 
5 AND  $\forall_{i=1}^m (V_i.vertexLabel = QV_i.vertexLabel)$ 
6 AND  $\forall_{j=1}^n (E_j.edgeLabel = QE_j.edgeLabel)$ 
7 AND  $\forall_{j=1}^n (E_j.sVertex = V_f.vertexID \text{ AND } E_j.dVertex = V_f.vertexID)$ ;

```

(TRANS-1)

Where each referenced table V_i (Line number 2) represents an instance of the table *Vertices* and maps the information of one vertex of the set of query vertices QV . Similarly, each referenced table E_j represents an instance of the table *Edges* and maps the information of one edge of the set of query edges QE . f is the mapping function between each vertex of QV and its associated vertices table instance V_i . Line number 3 represents a set of $m - 1$ conjunctive predicates to ensure that all queried vertices belongs to the same graph. Similarly, Line number 4 represents a set of n conjunctive predicates to ensure that all queried edges belongs to the same graph of the queried vertices. Lines number 5 and 6 represent the set of conjunctive predicates of the vertex and edge labels respectively. Line number 7 represents the topological edges connection information between the mapped vertices.

Verification phase: This phase is an *optional* phase. We apply the verification process only if more than one vertex of the set of query vertices QV have the same label. Therefore, in this case we verify that each vertex in the set of filtered vertices for each candidate graph database member is distinct. This can be easily achieved using their *vertex ID*. Although the fact that the checks of the verification process could be injected into the SQL translation template of the filtering phase, it is more efficient to avoid the cost of performing these conditions over each graph database members and we delay the processing cost of performing them (*if required*) to a separate phase after pruning the candidate list.

3.2 Edge-Edge Mapping Scheme

In the second proposed mapping scheme for *directed labelled graphs*, we start by assigning a unique identifier for each graph database member, vertex and edge in

the graph database. Each edge in the graph database is represented by a single tuple in the encoding relation. In addition to the ID of the graph member, each tuple stores the IDs and the labels of the source and destination vertices. The relational storage scheme of the *Edge-Edge* mapping is described as follows:

– *EdgeEdge*(graphID, edgeID, *edgeLabel*, *sVID*, *sVLabel*, *dVID*, *dVLabel*)

Figure 2 illustrates an example of the *Edge-Edge* relational mapping scheme of graph databases. In this figure, each underlined number in the representation of the directed labelled graph database members represents a *vertex ID* while each bold underlined symbol represents an *edge ID*. On one side, the *Edge-Edge* mapping scheme is more efficient for querying purposes. It groups the information of all vertices and edges into one *denormalized* relation. Therefore, assuming that we have a graph query q consists of m vertices and n edges. Using the *Vertex-Edge* mapping scheme, the number of join operations between the encoding relations is equal to $m + n$. However, using the *Edge-Edge* mapping scheme, it is reduced to only n join operations. On the other side, the *Vertex-Edge* mapping scheme is more suitable to deal with dynamic (with frequent updates) graph databases. It is more efficient to add, delete or update the structure of existing graph database members with *no* special processing. However, the *Edge-Edge* mapping scheme suffers from the update anomalies problem resulting from duplicating the label information of each vertex in the representing tuple of each edge it belongs to. Using the *Edge-Edge* mapping scheme, the SQL translation template of graph query q consists of a set of vertices QV with size equal m and a set of edges QE equal n is:

```

1 SELECT DISTINCT  $E_1.graphID, E_i.edgeID$ 
2 FROM EdgeEdge as  $E_1, \dots, EdgeEdge$  as  $E_n$ 
3 WHERE  $\forall_{i=2}^n (E_1.graphID = E_i.graphID)$ 
4 AND  $\forall_{j=1}^n (E_j.edgeLabel = QE_j.edgeLabel)$ 
5 AND  $\forall_{j=1}^n (E_j.sVertexLabel = QV_{sf(j)}.vertexLabel)$ 
6 AND  $\forall_{j=1}^n (E_j.dVertexLabel = QV_{df(j)}.vertexLabel)$ 
7 AND  $\exists_{x,y} (E_{g(x)}.sVertexID = E_{g(y)}.sVertexID)$ 
8 AND  $\exists_{x,y} (E_{g(x)}.dVertexID = E_{g(y)}.dVertexID)$ 
9 AND  $\exists_{x,y} (E_{g(x)}.sVertexID = E_{g(y)}.dVertexID)$ 

```

(TRANS-2)

Where each referenced table E_i (Line number 2) maps the information of one edge of the set of query edges. Line number 3 represents a set of $n - 1$ conjunctive predicates to ensure that all queried edges belong to the same graph member. Line number 4 represents the set of conjunctive predicates of the edge labels. Lines number 5 and 6 represent the set of conjunctive predicates of the labels of the source and destination vertices of each query edge respectively. sf and df are the mapping functions between the source and destination vertices of each query edge and their labels. Lines number 7, 8 and 9 represent the topological conditions of the graph query. Line number 7 represents a set of conjunctive predicates for each pair of query edges have the same source vertex. Line number 8 represents a set of conjunctive predicates for each pair of query edges have the

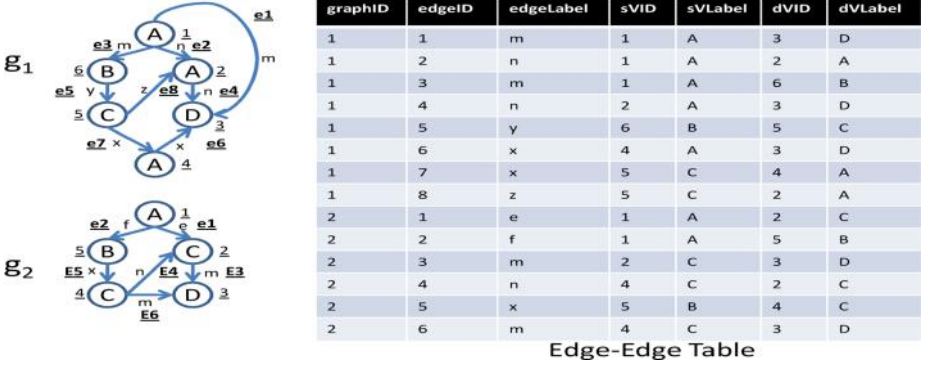


Fig. 2. Edge-Edge relational mapping scheme of directed label graphs

same destination vertex. Line number 9 represents a set of conjunctive predicates represents each pair of query edges where the source vertex of one edge is the destination vertex of another edge. g is the mapping function between each query edge and its associated table instance E_i .

In the *Edge-Edge* mapping scheme, the verification process is applied if more than one edge in the set of query edges QE have the same label. Therefore, we verify that each edge in the set of filtered edges for each candidate graph database member g_i is distinct. This can be easily achieved by comparing their *edgeID*.

4 Relational Processing of Undirected Labelled Graphs

4.1 Edge-Vertex Mapping Scheme

In the *undirected labelled graphs*, the direction information of each edge is not relevant. By default, each edge is *bidirectional*. Hence, there is no need to distinguish between the *source* vertex and the *destination* vertex of each edge in the encoding information. Instead, it is enough to specify that a vertex V belongs to an edge E . An intuitive *normalized* relational mapping scheme which can directly come to mind in the context of encoding undirected labelled graphs is to group the information (ID and label) of all vertices of the different graph database members into one (*Vertices*) relation, group the information (ID and label) of all edges into one (*Edges*) relation, and establish a *bridging relation* (*Edge-Vertex*) to store the information of the *many-to-many* relationships between the vertices and their associated edges (each vertex can belong to more than one edge and each edge is by default connects two vertices). Although the fact that this encoding scheme could be very efficient in terms of the storage space and the support of update operations. However, it is very expensive in terms of query operations. Let us assume that we have a graph query q consists of m vertices and n edges. Using this *normalized* mapping scheme, we will need

to join between m instances of the (*Vertices*) relation, n instances of the (*Edges*) relation and $2n$ instances of the bridging relation (*Edge-Vertex*). In addition, it will require $2n + m$ conjunctive predicates to ensure that all vertices, edges and bridging records belong to the same graph database member. Therefore, it is expected that most of relational query engines will certainly fail to execute the SQL translation queries of medium size or large graph queries because they are too *long* and too *complex*. To tackle this problem, we encode undirected labelled graphs using a *denormalized* single relation *EdgeVertex*. In this relation, each edge is represented by two tuples (one for each connected vertex). Each tuple stores the graph database member which the edge belongs to, the edge label and the information of one of the connected vertices. The schema of this *Edge-Vertex* encoding relation is described as follows:

- *EdgeVertex*(*graphID*, *edgeID*, *edgeLabel*, *vertexID*, *vertexLabel*)

A clear advantage of this *denormalized* relation is that the SQL evaluation of the graph query will need to only join $2n$ instances of the encoding relation *EdgeVertex* instead of $3n + m$ in the case of using the *normalized* scheme. Consequently, the number of the required conjunctive predicates to ensure that all queried information belongs to the same graph database member is reduced to $2n$ instead of $3n + m$ in the *normalized* scheme. Figure 3 illustrates an example of the Edge-Edge mapping scheme of graph databases. Using the *Edge-Vertex* mapping scheme, the SQL translation template of a graph query q consists of a set of vertices QV with size equal m and a set of edges QE equal n is:

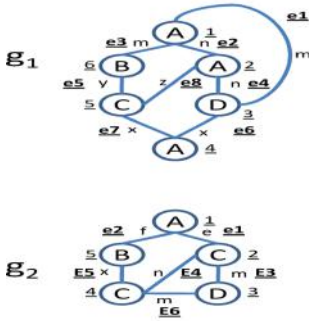
```

1 SELECT DISTINCT  $E_1.graphID, E_i.vertexID$ 
2 FROM EdgeVertex as  $E_1, \dots, EdgeVertex$  as  $E_{2n}$ 
3 WHERE  $\forall_{i=2}^{2n} (E_1.graphID = E_i.graphID)$ 
4 AND  $\forall_{j=1}^m (E_{f1(j)}.edgeID = E_{f2(j)}.edgeID)$ 
5 AND  $\forall_{j=1}^m (E_j.edgeLabel = QE_j.edgeLabel)$ 
6 AND  $\forall_{j=1}^{2n} (E_j.vertexLabel = QV_{f3(j)}.vertexLabel)$ 
7 AND  $\exists_{x,y} (E_{g(x)}.vertexID = E_{g(y)}.vertexID)$ 

```

(TRANS-3)

Where each referenced table E_i (Line number 2) maps the information of one of the queried edge-vertex connections. Line number 3 represents a set of $2n$ conjunctive predicates to ensure that all queried edge-vertex connection information belong to the same graph member. Line number 4 represents a set of m conjunctive predicates to ensure that every two table instances of the same query edge represent the same *edgeID*. $f1$ and $f2$ are the mapping functions between each query edge and its two distinct table instances. Lines number 5 and 6 represent the set of conjunctive predicates for filtering the labels of the queried edges and vertices respectively. $f3$ is the mappings function between each query edge and a distinct vertex of its two connected vertices. Line number 7 represents a set of conjunctive predicates which are describing the topological structure of the graph query. Each predicate describes a pair of query edges which is sharing one of the queried vertices.



Edge-Vertex Table

Fig. 3. Edge-Vertex relational mapping scheme of directed label graphs

In the *Edge-Vertex* mapping scheme, the verification process is applied if any of the query edges connects between two vertices with the same label. We then verify that the resulting edge does not connect between two vertices with the same *vertexID*.

5 Relational-Based Optimization for Graph Queries

Clearly, an obvious bottleneck of the SQL-based evaluation of the graph queries is that it involves a large number of conjunctive SQL predicates and join operations. Relational query optimizers derive much of their performance from using of physical properties that are specific to the relational model such as: sortedness, proper join ordering and powerful indexing mechanisms. In this section we will explain our approach to employ the powerful features of the relational query engines and their efficient indexing schemes to achieve efficient relational execution plans for our queries.

5.1 Injecting Selectivity Annotations for SQL Predicates

In general, one of the most effective techniques for optimizing the execution times of SQL queries is to select the relational execution plan based on accurate selectivity information of the query predicates. For any given SQL query, there is a large number of alternative execution plans. These alternative execution plans may differ significantly in their use of system resources or response time. For example, in our SQL queries, the query optimizer may need to estimate the selectivities of the occurrences of the two vertices in one subgraph, one of these vertices with label *A* and the other with label *B* in order to choose the more selective vertex to be filtered first. Providing an accurate estimation for the selectivity of the predicates defined in our SQL translation templates requires having



Fig. 4. Sample Markov tables summaries of Vertex-Edge mapping

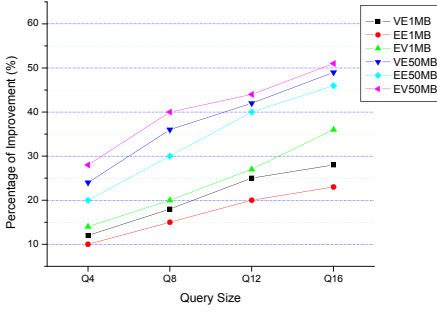
statistics that contain information about the structure of the stored graph data. These statistics must be small enough to be processed efficiently in the short time available for query optimization and without any disk accesses. Therefore, we construct three Markov tables to store information about the frequency of occurrence of the distinct labels of vertices, distinct labels of edges and connection between pair of vertices (edges). Figure 4 presents an example of our Markov table summary information. In our context, we are only interested in label and edge information with low frequency. Therefore, it is not necessary and not useful to keep all such frequency information. Hence, we summarize these Markov tables by deleting high-frequency tuples up to a certain defined threshold *freq*. These information about the low frequency labels and edges are then used to effectively prune the search space, reduce the size of intermediate results and influence the decision of the relational query optimizers to select the most efficient join order and the cheapest execution plan. We use the statistical summary information to give influencing hints for the query optimizers by *injecting* the selectivity information for the individual query predicates into the SQL translations of the graph queries. These hints enable the query optimizers to decide the optimal join order, utilizing the most useful indexes and select the cheapest execution plan. In our experiments, we used the following syntax to pass the selectivity information to the IBM DB2 query optimizer:

```
SELECT fieldlist FROM tablelist WHERE  $P_i$  SELECTIVITY  $S_i$ 
```

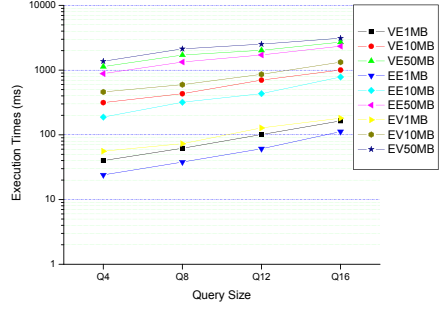
Where S_i indicates the *selectivity* value for the query predicate P_i . These selectivity values are ranging between 0 and 1. Lower selectivity values (close to 0) will inform the query optimizer that the associated predicates will effectively prune the number of the intermediate result and thus they should be executed first.

5.2 Relational Indexes Support for SQL-based Graph Queries

Relational database indexes have proven to be very efficient tools to speed up the performance of evaluating the SQL queries. Moreover, the performance of queries evaluation in relational database systems is very sensitive to the defined indexes structures over the data of the source tables. We use *partitioned B-tree indexes* as a *standard*, powerful and matured indexing mechanism to accelerate



(a) Optimization of Graph Queries



(b) Performance Scalability

Fig. 5. Performance characteristics of relational evaluation of graph queries

the performance of the SQL evaluation of the graph queries. Partitioned B-tree indexes are considered to be slight variant of the B-tree indexing structure. The main idea of this indexing technique has been presented by Graefe in [11] where he recommended using of low-selectivity leading columns to maintain the partitions within the associated B-tree. For example, in labelled graphs, it is generally the case that the number of *distinct* vertices and edges labels are far less than the number of vertices and edges respectively. Hence, for example having an index defined in terms of columns (*vertexLabel*, *graphID*) can reduce the access cost of the graph query with only one label to one disk page which is storing a list of *graphID* of all graphs which are including a vertex with the target query label. On the contrary, an index defined in terms of the two columns (*graphID*, *vertexLabel*) requires scanning a large number of disk pages to get the same list of targeted graphs. Conceptually, this approach could be considered as horizontal partitioning of the encoding relations using the high selectivity attributes. Therefore, instead of requiring an execution time which is linear with the number of graph database members, Having partitioned B-trees indexes of these high-selectivity attributes can achieve fixed execution times which is no longer dependent of the size of the graph database [11][12].

6 Performance Evaluation

In this section, we evaluate the performance of our proposed relational-based techniques for storing and querying various kinds of graph data. We carried out a series of performance experiments in order to study the tradeoffs of the alternative proposed schemes. We conducted our experiments using the IBM DB2 RDBMS running on a PC with 2.8 GHZ Intel Xeon processors and 4 GB of main memory. In our experiments we use the DBLP dataset which presents the database of bibliographic information of computer science journals and conference proceedings [1]. We converted the available XML tree into directed labelled graphs by using edges to represent the relationship between different entities of

the datasets such as: the ID/IDREF, cross reference and citation relationships. Four query sets are used, each of which has 1000 queries. These 1000 queries are constructed by randomly selecting 1000 graphs and then extracting a connected m edges subgraph from each graph randomly. Each query set is denoted by its edge size as Q_m . In principle, our experiments have the following goals: 1) To show the efficiency of the proposed techniques in terms of their execution times for graph queries and their scalability to deal with very large data sets and large subgraph queries. 2) To show the efficiency of our optimization techniques to improve the performance of our proposed SQL-based evaluations of the graph queries.

Experiment I: Relational Optimization of Graph Queries. Figure 5(a) indicates the average percentage of speed-up improvement on the execution times of the SQL-based evaluation of the graph queries using the partitioned B-tree indexing technique and the injected selectivity annotations. In these experiments we used two instances of the DBLP dataset. One instance of size 1 MB and the other of size 50 MB. For each instance, we used query groups with different edge sizes of 4, 8, 12 and 16. The generated queries are evaluated using each mapping scheme. We use the symbols VE , EE , EV for referring to the *Vertex-Edge*, *Edge-Edge* and *Edge-Vertex* mapping schemes respectively. The reported percentages of speed up improvements are computed using the formula: $(1 - \frac{G}{C}) \%$ where G represents the execution times of the SQL execution plans using the partitioned B-tree indexes and the injected selectivity annotations and C represents the execution time of the SQL execution plans using the traditional B-tree indexes and without the injected selectivity annotations of the SQL predicates. The results of these experiments confirm the efficiency of both optimization techniques on improving the SQL-based evaluation of graph queries. Comparing the alternative mapping schemes, we can see that the percentage of improvement of the *Edge-Vertex* is the highest. The main reason behind this is the it performs join operations between larger encoding tables. These joins between the large tables are more expensive specially when most of the data in the joined relations are irrelevant for the graph queries. Therefore, using the selectivity annotations and the partitioned B-tree indexes play an effective role on improving the execution times of these queries by dramatically filtering and reducing the access cost of the *relevant* records. Clearly, the bigger the query size, the more join operations are required to be executed on all mapping scheme and the higher the improvement of both optimization techniques.

Experiment II: Performance Scalability. One of the main advantages of using RDBMSs to store and query graph databases is to exploit their well-know scalability feature. To demonstrate the scalability of our approach, we conducted a set of experiments using different sizes of the DBLP dataset: 1, 10 and 50MB. Figure 5(b) illustrates the average execution times for the SQL-based evaluation of the 1000 graph queries. The results of these experiments confirm the high scalability of our approach. Obviously, the results of Figure 5(b) shows that the execution times of the proposed mapping schemes scale in a near linear fashion

with respect to the graph database and the query size. The main reason behind this is the power and the efficiency of relational query engines on dealing with relatively large and complex queries which may involve large number of joins and filtering conditions. Comparing the alternative mapping schemes, we can see that the *denormalized Edge-Edge* mapping scheme wins over the *normalized Vertex-Edge* mapping scheme for processing graph queries over *directed labelled graphs*. This result is quite expected and can be explained fairly easily because of the effective reduction on the required number of expensive join operations between the encoding relations. The results of the *Edge-Vertex* scheme for processing graph queries over the *undirected labelled graphs* show that it has the longest execution times. The main reason behind this is the cost of join operations between the large encoding tables. However, these longest execution times are still getting the most benefit from our optimizing techniques (Figure 5(a)) to get rid of the irrelevant data very early.

7 Conclusions

With the growing importance of using graphs for modelling complicated data, there is a crucial need on devising new techniques for processing graph queries. Our focus on this paper has been to study the virtues and limitations of exploiting the well-known maturity of the relational infrastructure for achieving efficient performance of processing graph queries. The potential advantages of this approach are: reusing existing mature technology and using an existing high performance systems. Therefore, it can reside on any relational database system and exploits its well known matured query optimization techniques as well as its efficient and scalable query processing techniques. In addition, our approach does not require any extra time cost for offline or pre-processing steps. Our experiments have confirmed that the efficient performance of our approach and its ability to handle large sizes of graph databases and large graph queries. Therefore, we believe that it deserves to be pursued further. In the future, we will study the feasibility of extending our approach to deal with other types of graph queries such as similarity and approximate queries. In addition, we are planning to investigate the possibility of employing other relational-based optimization techniques such as materialized views on improving the execution times of the SQL-based evaluation of graph queries.

References

1. DBLP XML Records, <http://dblp.uni-trier.de/xml/>
2. Abounnaga, A., Alameldeen, A., Naughton, J.: Estimating the Selectivity of XML Path Expressions for Internet Scale Applications. In: VLDB (2001)
3. Zhang, S., et al.: TreePi: A Novel Graph Indexing Method. In: ICDE (2007)
4. Cai, D., Shao, Z., He, X., Yan, X., Han, J.: Community Mining from Multi-relational Networks. In: Jorge, A.M., Torgo, L., Brazdil, P.B., Camacho, R., Gama, J. (eds.) PKDD 2005. LNCS (LNAI), vol. 3721, pp. 445–452. Springer, Heidelberg (2005)

5. Huan, J., et al.: Mining protein family specific residue packing patterns from protein structure graphs. In: *Computational Molecular Biology* (2004)
6. Yoshikawa, M., et al.: XRel: a path-based approach to storage and retrieval of XML documents using relational databases. *TOIT* 1(1) (2001)
7. Zhao, P., et al.: Graph indexing: tree + delta = graph. In: *VLDB* (2007)
8. Cohen, S., et al.: Scientific formats for object-relational database systems: a study of suitability and performance. *SIGMOD Record* 35(2) (2006)
9. Botea, V., et al.: PIST: An Efficient and Practical Indexing Technique for Historical Spatio-Temporal Point Data. *GeoInformatica* 12(2) (2008)
10. Giugno, R., Shasha, D.: GraphGrep: A Fast and Universal Method for Querying Graphs. In: *International Conference in Pattern recognition* (2002)
11. Graefe, G.: Sorting And Indexing With Partitioned B-Trees. In: *CIDR* (2003)
12. Grust, T., Rittinger, J., Teubner, J.: Why Off-The-Shelf RDBMSs are Better at XPath Than You Might Expect. In: *SIGMOD* (2007)
13. Grust, T., Sakr, S., Teubner, J.: XQuery on SQL Hosts. In: *VLDB* (2004)
14. Klinger, S., Austin, J.: Chemical similarity searching using a neural graph matcher. In: *European Symposium on Artificial Neural Networks* (2005)
15. Yan, X., Yu, P., Han, J.: Graph indexing: a frequent structure-based approach. In: *SIGMOD* (2004)
16. Zhang, N., Özsu, T., Ilyas, I., Aboulmaga, A.: FIX: Feature-based Indexing Technique for XML Documents. In: *VLDB* (2006)
17. Zou, L., Chen, L., Xu Yu, J., Lu, Y.: GString: A novel spectral coding in a large graph database. In: *EDBT* (2008)

SecCom: A Prototype for Integrating Security-Aware Components

Khaled M. Khan¹ and Calvin Tan²

¹ Department of Computer Science and Engineering, Qatar University, Qatar
k.khan@qu.edu.qa

² School of Computing and Mathematics, University of Western Sydney, Australia

Abstract. This paper addresses information security from systems development point of view. This paper presents a prototype demonstrating how security-aware software components can be composed with other remote objects in terms of security compliances. It shows that the integration between two third-party components can be formed based on the compliance of their security requirements and assurances. With a running example, the paper attempts to demonstrate how the compliance of security requirements of a component is checked, and how a viable integration between software components is formed matching the security requirements of each other. The paper also describes the underlying architecture of the prototype.

Keywords: Software components, security-aware, composition, security properties, software services.

1 Introduction

With the introduction of service software such as Web services, information systems are increasingly becoming outsourced in terms of their composition and execution. The concept of component based software system is based on the readily and openly available internet protocols, and easier framework for software integrators to adopt. An information system can be integrated with other software services/components readily available from various distributed sources for run-time integration and execution. In a service oriented software environment, an integration process consists of multiple autonomous geographically dispersed software components with no shared memory [1]. A software component offering a service to other software systems is autonomous as it has its own executable code, and uses its own data or files. The use of software service is attractive to system developers, because the paradigm supports reusability of code, and far efficient utilization of systems resources [6].

However, the use of third party software service needs to address the security more carefully than the case for the traditional stand-alone systems security. When software components are acquired from distributed sources, and integrated with an information system, the security impact of such components needs to be

known a priori and the security properties should be verified before a viable composition is established. In a service-centric environment which is highly fluid and unpredictable, security properties of third-party software components located in remote servers may be unknown to the developer of the information system. The security properties of a software component in isolation may be different from the compositional impact of the same component within an information system [5].

In a highly fluid environment, the claimed security properties of third party software components should be adequately specified. These properties need to be readily available and verifiable during the component integration process. There are technologies such as JavaBeans, DCOM, OMG's CORBA, WS-Security for effective composition. What is lacking in those technologies is a mechanism that can be used by the developers to check the compositional security properties of the participating components. Existing component technologies provide interface signatures for effective structural communication among components [2]. The interface signatures do not provide any mechanisms to reason about the expected security behavior and the post composition effect of the system. Security policies of software components are expressed in various formats and levels of granularity [7].

In current practices, the service integration is formed without checking the potential security compromises between participating components. The practice virtually forces the service consumers to make an integration for a service by ignoring the security issues related to the integration [4]. This is unacceptable especially for service-oriented systems which are usually deployed in the open hostile environment. Most of the models use text-based checklists which do not have any reasoning capabilities. Thus, they lack a common representation of security properties for compositional purposes.

To make security integration viable between components, the security properties of the participating components need to be checked in their use context for the security compliance. The use of service software provides a mix of business benefits and risks. Addressing those risks takes a coordinated effort between executives, management, and operations personnel to establish a common strategy, and define tactics [10]. The challenge is to formulate a uniform model of software components that exposes security properties for interactions with others, and provides reasonable mechanisms to verify their compliances with others.

Some research activities related to this area have reported some promising results. The work published in [7] defines a framework for expressing security certification model (SCM). The framework is intended to address the security of software components [7]. A predictive model for identifying software components prone to failure during security attacks has been proposed in [8]. The use of user system interaction effect (USIE) model [9] to derive and analyze security concerns from service-oriented software architectures is a promising technique. It may requires further development to make the model more automatic.

To augment the research on run-time security compliance checking between information systems and third party software services further, we develop a

simple prototype. The tool is based on the principles of our work reported in [4,5]. The next section outlines the main concepts of security-aware composition. The prototype is presented in section 3 with an example. The applicability of the prototype is demonstrated with the example. Section 4 describes the architecture of the prototype. The paper concludes in section 5. We use software component and software service interchangeably.

2 Security-Aware Integration

In service oriented software systems, a software component provides a computational service to other information systems. The system that uses the services of another software component is called a *focal* system. The software component that provides the requested service to a focal system is called *candidate* system [3]. Security properties of software systems (focal and candidate systems) are associated with the corresponding functionality that a software component provides to others, or receives from other systems. Security properties of a system are grouped into two classes: required security properties and ensured security properties [5]. The required properties are those which must be satisfied by other systems in order to be integrated for a specific service. The ensured properties are those which are guaranteed by the component to others.

The success of an integration between an information system and a software component depends on security requirements and assurances of the participants. In order to form a viable integration, compatibility checking of security properties between the participating components is required. The comparison between the required security properties of a system and the ensured security properties of another system is called security compliance checking. The result of a security compliance checking produces a *compositional security contract (CsC)* if it is successful [4]. An information system can use a third party service if the service satisfies its security properties. Similarly, in delivering a service and assuming the required security properties are satisfied for that service by the focal component, the software service ensures certain security properties. A security-aware integration is based on the principles of checking compatibility between the security properties of two participating systems [4].

In order to express and verify the security properties, we have chosen a declarative notation which is based on logic programming [4]. In principle, component developers could use a variety of languages to express security properties. The language could be a propositional or constraint language. For instance, the security properties could be written in XML for storage purposes. The simple structure of logic program allows us to represent complicated form of security knowledge and reasoning. In logic programming, security properties are represented and expressed in symbolic notations called *atoms*. An *atom* consists of a predicate name (a security function) with variables or constants (elements). An element can represent an entity, data, a password, a key etc. Variables are used to generalize objects such as X for which the inference rules of logic programming find or substitute an element.

The predicate name of an atom represents a security property such as *encrypted*, *key_generated* and *digitally_signed* etc. An example of an atom is *digitally_signed(p,x)*. It states that an element identified as p digitally signs the object x . The entity p could be any entity such as a software, a person, a component. The object x could be a file, a data, or a message. In addition to the representation of security properties as atoms, inference rules in logic programming are used to check the conformity of the security properties represented in atoms. Rules make an inference with the knowledge associated with the security properties codified in atoms. If each atom in the body of a rule is *true*, then the inference rule concludes that the properties satisfy compliances. Rules make an inference with the security properties of the participating components. A set of inference rules are applied to prove whether a CsC is achievable or not.

To clarify the concepts further, consider a simple service oriented software system that caters the following application to users: searching an element in a huge database which stores highly classified data. The application involves three services: *generating random numbers*, *sorting the randomized numbers*, and *searching an element in the numbers*. Several third-party components residing in different remote machines provide these services such as generate huge number of randomized elements, sorting elements in certain order, and searching an element. We assume some randomizer components developed by different companies produce randomized numbers for their clients. Each of these components has their own security requirements and assurances although their services are identical, that is, generate random numbers. Similarly, there are many independent software components providing sorting services, and some offer searching services to other systems.

Assume that the required security property of a randomize component states that a request for random elements and the type of elements provided by a client software must be encrypted with its public key. Whereas it ensures that all generated randomized numbers are digitally signed by the randomizer component. The inference rules of logic programming reason about the properties if the elements are encrypted or not by the client component, and the resulted randomized elements are digitally signed by the components or not. A CsC is formed between the randomizer component and its client system if these properties are satisfied. The CsC in this case would be digitally signed randomized numbers. The client then sends the digitally signed random numbers to another component which sorts the elements in certain order. The sorting component may have its own security requirements. If the requirements are met, then it sorts the data and returns these to the client with the security assurances if there are any. Based on this scenario, we demonstrate our prototype called SecCom in the next section.

3 SecCom Prototype

The architecture of the SecCom is based on several processing units. Section 4 outlines the architecture of the prototype. One of the classes used in the prototype is *SecComProvider* which provides the participating software components

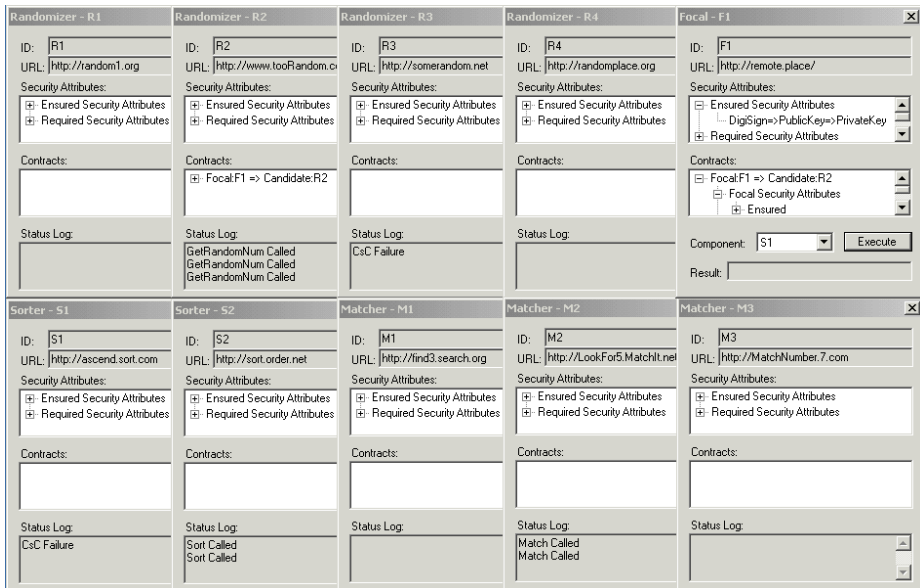


Fig. 1. The focal component and candidate components in the SecCom prototype

the capability to test the security compliances. When a focal component initiates an integration with a candidate component, a security contract is attempted to be established between the two participants. This involves the components checking to see if the required security properties of one component can be met by the corresponding ensured properties of the other. If the supporting security attributes of a system satisfy the requirements, a compositional security contract (CsC) is formed between the two components. The CsC is then stored in both components. Once the integration is complete, the candidate component allows the focal component to execute its service.

The focal component (representing an information system or business object) initiates the interaction with the other software components using the class *SecComProvider* class. Each software component is instantiated, and runs as a separate component in its own window. Each window displays information about the component, and reports any interaction that has occurred.

3.1 Processing Example

A focal component identified as *F1* is used to instantiate other components as individual objects. Assume that we have identified and loaded the following components in our prototype: one *focal* component *F1*; four *candidate* randomizer components identified as *R1*, *R2*, *R3* and *R4*; two *candidate* sorter components such as *S1* and *S2*; and three *candidate* matcher objects identified as *M1*, *M2*, and *M3*. Each of these components has their own required and ensured security

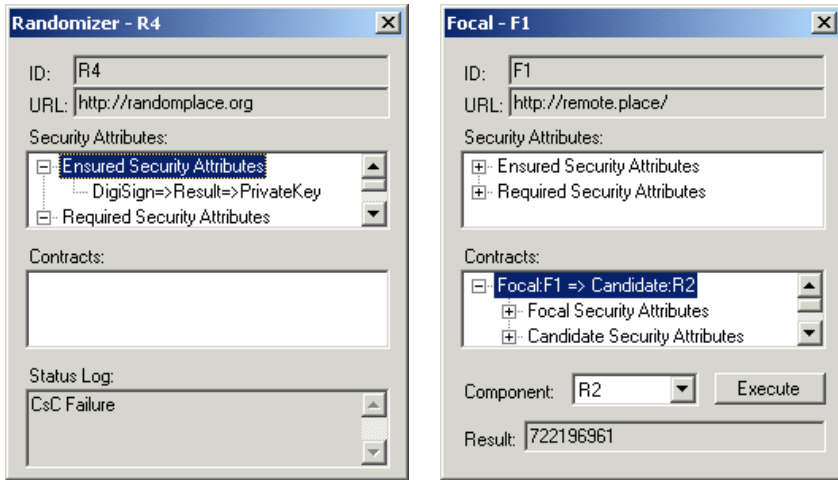


Fig. 2. An example of interfaces of two components in SecCom

properties. In our prototype, a component is represented with a separate window as shown in Fig. 1. The figure depicts all candidate components and the focal component.

Each of these individual components has unique name, URL location and security attributes. The components have been written and compiled as class library files (*.dll) with the exception of the focal component which is an executable. Fig. 2 shows the interfaces of the component *R4* (randomizer) and *F1* (focal component) as examples. Notice that the component *R4* has an *ID*, its originating *URL*, its security properties (*required* and *ensured*). The ensured properties state that it provides digitally signed random elements to its client, and it signs the elements with its private key. In the *Contracts* field, it stores the CsC (agreed security properties) with its client if there is any. The *Status Log* field keeps track the result of the verification of the security compliance checking such as non compliances. The focal component *F1* has similar first four data fields similar to *R4*, but it has an additional field called *Component*. This field has a scroll down menu showing all the components that *F1* has loaded, and tries to test their suitability in regards to the security requirements of *F1*. This is done by choosing the component in the drop-down list and clicking on the execute button. The *Contracts* field shows the current data '*Focal F1 = Candidate : R2*'. It means that the focal component *F1* has made a contract with the candidate component *R2*. The security properties of both components are also stored in this field. The CsC can be referred to other components if required.

The prototype has three individual components of randomize type providing same functionality. Each of these components generates a series of randomized elements of specific type for their clients. Each of them has specific required as well as ensured security properties. The service is provided if its security requirements are met by the client, that is, *F1*. We assume that a CsC has

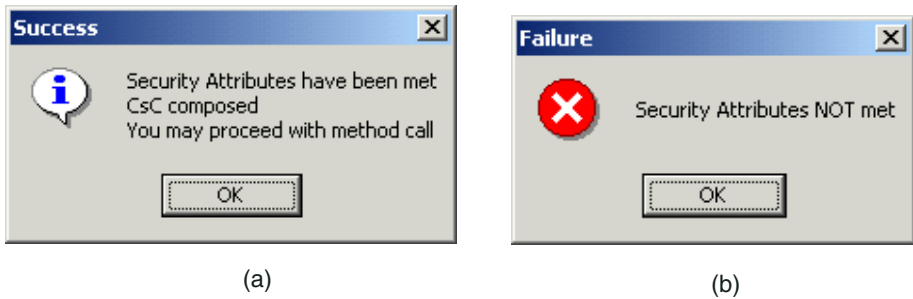


Fig. 3. Examples of two possible results of the checking

been formed between $F1$ and $R2$. The results obtained from the service, after clicking on the *Execute* button as shown in $F1$ (see Fig. 2), are displayed in the Result box. In this case the result generated by the component $R2$ is the random elements 722196961 stored in the *Result* field as shown in Fig. 2.

The sorter is the second type of component in this example. Each instance of this component type provides sorting functionality to their client. The sorter component accepts a series of huge number of random elements as an argument from its client, and returns sorted elements in ascending order. If the random numbers do not include the digital signature of a randomizer component, it simply returns an empty string. That means, the digital signature of the random numbers generator is the required security property of the sorting component. The *sort()* method of this component is only accessible by the clients if its security requirement is ensured. In our example, the CsC between $F1$ and $R2$ has been formed based on the digital signature of $R2$. In this case the required properties of the *sort()* component $S2$ match this. So a second CsC has been formed between $F1$ and $S2$.

The matcher component type contains the *match()* method. Its job is to search for a specified element in the series of sorted random numbers obtained from $F1$. It returns a message '*Found*' or '*NotFound*' based on the result of the processing as shown in Fig. 3 (a) and (b) respectively. The random number sequence is passed to the *match()* method as an argument either sorted or not. We assume that the security compliance between the *match()* component $M2$ and the focal component $F1$ is done. A CsC is then established between $F1$ and $M2$. All formed security contracts are added to the *Contracts* field of the component. In our example, a software application is composed based on three components and a focal system: $F1$, $R2$, $S2$, and $M2$. This composed system provides a functionality: searching an element in a huge database which stores highly classified data generated randomly. The system is considered security complied because the security requirements of all its three components have been satisfied. If the focal component attempts to make an integration with a candidate component for which a security contract has already been established, the process of checking security compliance does not execute as depicted

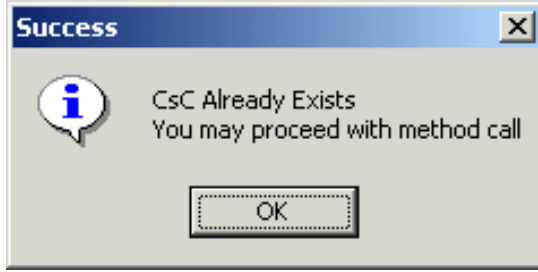


Fig. 4. Previously established contract

in Fig. 4. Instead, the service of the component is allowed to be executed by the focal component. The function identifies the exiting CsC and prevents creating duplicate CsC between same components.

3.2 Running of SecCom.

The program starts by running the executable program *focal.exe*. This will open ten windows representing all components as already shown in Fig. 1. All operations of SecCom can only be performed using the focal component *F1* by choosing a candidate component. When an interaction occurs between two components, the progress of the operation is reported in the corresponding windows of the component. To keep the prototype simple, all details and attributes of the components have been hard-coded. The security properties of the components are fixed. In our example, components such as *R2*, *S2*, *M2* have security attributes that actually meet the necessary security requirements of the focal component *F1*. Selecting any of these three components based on the security requirements of the focal component results in either a successful security contract or a CsC already exists. Choosing any other components results in a failure in security compliance. All data resulted in from operations in SecCom are stored in the *Status Log* field of the participating components.

4 Implementation

The basic architecture of the prototype supports two stages as shown in Fig. 5. The reason for the two-stage approach is to keep the preprocessing of input information independent from the transforming the security properties into well formed output. In some cases, the compositional process may not perform the result generation stage immediately, or there may be other external tasks to be completed before generating the compositional result. This two-stage approach enables the system to carry out the first stage, namely pre-processing and leave the second stage, transformation, to the time when the system requires the output. The architecture of the prototype has two major processing components:

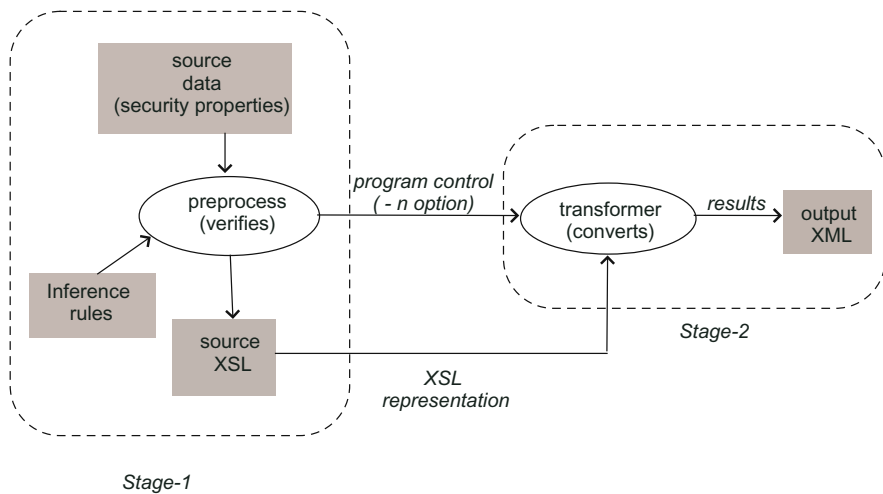


Fig. 5. Architecture of the prototype

pre-processor(verify) and *transformer(converts)* as shown in Fig. 5. The tool has also four different data files: *source data (security properties)*, *inference rules*, *source XSL*, and *output XML*. The pre-processor derives the security properties of the participating components from their source data files, and the inference rules. It then produces a source XSL representation of the security properties and the rules. The transformer component gets the input data from the source XSL, makes the reasoning based on the inference rules, and generates output in XML format. The processing of transformer is quite complicated because it involves reasoning about the security properties of the participating components. The output stores the results of the security compliance such as forming CsC.

The inference rules are considered the set of reasoning protocols. Security concerns is also a reason not to keep users' data in a file on the system. The processing components use java implementations of XML and XSLT parsers. The pre-processor has the task of reading the security properties and inference rules, and generating an XSL file from the processing. It then passes the XSL file together with the security properties used to the transformer. The transformer merely produces the CsC results and stores them in the log and contract fields of the corresponding components.

XML has been selected in the prototype for its easy readability. The source data is a data stream that contains data in a structured format. An XML document contains data in structured format which can easily be read or processed later. The choice of XML has been decided upon due to the nature of XML itself being a platform for communicating data. The modern database systems support retrieving data in XML format. SecCom makes use of XML as means of communicating the data and a small Java program to conduct the verification process. Although the sample application uses XML (plain text) as source

(containing security properties of the component) and verification data (the rules to make the matching), this model suggests that it may be extended to other types of data, including objects in the future. The data verification does not include checking of data types or data ranges, most programming and scripting languages already provide features to perform these functions.

The engine that drives the verification process (including a simple XML/XSLT parser) has been written in Java. Some other alternatives were examined, however, for the purpose of this paper, the XML and Java combination has been chosen taking into account common standards, cross platform capabilities, suitability and availability. The alternatives considered, but is not limited to, includes making use of XML/DTD, XML Schemas, Document Object Model (DOM), Visual Basic Scripting. There are many other techniques of implementing this model and actual implementation method should depend on the environment of the target platform. Based on the above arguments, this tool has used the XML and Java combination framework.

5 Conclusion

This paper has presented a prototype with an example to argue that security-aware software composition is a viable option in service oriented systems. As the software services are independently developed by third-party in order to provide computational services to other software systems, the compliance of their security properties is an important issue to get credible security assessment for the service composition. The underlying ideas presented here could be further extended to all types of distributed software systems which require inter connectivity with other systems. We have dealt the issue of security from a software development perspective. We are now working to develop a security-aware composition shell which could be used as a platform by independent components to check the security properties of other. Our shell will act also as a software service for others.

References

1. Arafteh, B.: A Graph Grammar Model for Concurrent and Distributed Software Specification-in-Large. *Journal of Systems Software* 31, 7–32 (1995)
2. D'Souza, D., Wills, A.: *Objects, Components, and Frameworks with UML - The Catalysis Approach*. Addison-Wesley, Reading (1998)
3. Han., J.: A Comprehensive Interface Definition Framework for Software Components. In: *Proc of 1998 Asia-Pacific Software Engineering Conf.*, Taipei, Taiwan, December 1998, pp. 110–117 (1998)
4. Khan, K., Han, J.: A Security Characterisation Framework for Trustworthy Component Based Software Systems. In: *Proc of the 27th Annual Int'l Computer Software and Applications Conf. (COMPSAC 2003)*, Dallas, pp. 164–169 (2003)
5. Khan, K., Han, J.: Composing Security-Aware Software. *IEEE Software*, 34–41 (January/February 2002)

6. Pandey, R., Hashii, B.: Providing Fine-Grained Access Control for Mobile Programs Through Binar. In: Guerraoui, R. (ed.) ECOOP 1999. LNCS, vol. 1628, pp. 449–473. Springer, Heidelberg (1999)
7. Kelkar, M.: Modeling Software Component Security Policies. Doctoral thesis, University of Tulsa, Tulsa, OK, USA (2007)
8. Gegick, M., Williams, L., Vouk, M.: Predictive Models for Identifying Software Components Prone to Failure During Security Attacks. In: Proceedings Conference of OOPSLA, Nashville, Tennessee (October 2008)
9. Liu, Y., Traore, I.: Systematic Security Analysis for Service-Oriented Software Architectures. In: Proceedings of the IEEE International Conference on e-Business Engineering, pp. 612–621. IEEE Computer Society press, Los Alamitos (2007)
10. Willett, K.: Security Issues in Service-Oriented Architecture, CSC Online World (January/March 2007), <http://www.csc.com/cscworld/012007/fa/fa005.shtml>

Incorporating Software Testing as a Discipline in Curriculum of Computing Courses

Simi (Kamini) Bajaj¹ and Shyamala Balram²

¹ University of Western Sydney, Locked Bag 1797, Penrith South DC, 1797, NSW, Australia
Tel: +61-2-96859253, Fax: +61-2-96859557

k.bajaj@uws.edu.au

² Vice President & Head - Sydney PACE Lab, Polaris Software Pty Limited. Level 9, 31
Market Street, Sydney NSW 2000

Visiting Lecturer, UWS

shyamala.balram@polaris.co.in

Abstract. One of the very important aspects of software quality is software testing. Software testing consumes 30%-50% of the most software projects. Testing should not only be performed for the quality or development process requirement but due to the fact that testing needs to ensure that the software performs what it is supposed to do. UWS has responded to this need of the software testing in industry by introducing a course on Software Testing- 'Fundamentals of Software Testing' in undergraduate studies. This course is an outcome of strategic alignment of UWS with Polaris Software Pty Ltd involved in testing of large software projects. The course aims to prepare students for understanding of Software Testing Life cycle. This course is jointly delivered by academics and software testing specialists from industry, to provide a holistic view of software testing as practiced by the software industry.

Keywords: Software Testing, Software testing education, testing training, Test management.

1 Introduction

According to [1], 70% of software applications are built with defects, exceeded cost and time. There is enough evidence on the chaotic state to software development ranging from studies like The Robbins-Gioia Survey (2001), The Conference Board Survey (2001), The KPMG Survey (1997), The Chaos Report (1995), The OASIG Survey (1995) which collectively identify a failure rate of software development projects 50-70% [2]. There are research groups working on identifying the status such as Standish group, Gartner Research, and Forrester research. [3] reports money spent on cancelled IT projects in United Kingdom in last five years was AU\$664 million. There are many examples where organizations failed to test the software such as Patriotic Missile Defense System (1991), Disney Lion King (1994-95), Intel Pentium Floating Division Bug (1994), NASA Mars Polar Lander (1999), Dangerous viewing ahead (2004) [4]. All these examples clearly indicate a gap in the software development processes which fail to develop applications with acceptable level of defects.

Software testing is a process of *verifying* and *validating* that a software application or program a) Meets the business and technical requirements that guided its design and development, and B) works as expected. Software testing also identifies important *defects*, flaws, or errors in the application code that must be fixed. It's an investigation conducted to provide information about the quality of the software (Wikipedia website). Hence, software testing has three main purposes: verification, validation, and defect finding. It is a time honored approach for evaluating the software in terms of correctness, robustness, efficiency, functionality and ease of use [5].

Bill Gates [6] stated that there is one tester for every developer [7] and testers spend all their time testing and developers spend half their time testing. Australia's ICT market is around \$89 billion and out of that the testing market size is over \$4 billion¹. The demand for number of testers is over 50000. Apart from Australia there are huge opportunities available for software testers worldwide. There are approx. 1500 jobs advertised for Testers on Seek (Australian Jobs Website) every month.

To address this growing demand for testers, many initiatives are being taken. The International Software Testing Quality Board (ISTQB) provides internationally recognized professional qualification in software testing. The government has taken initiatives by developing Australian and New Zealand testing Board (ANZTB). Many universities in Australia have started offering courses in Software Testing with a few of them in Sydney. In April 2008, the Vice Chancellor, University of Western Sydney and Head of Sydney Operations, Polaris Software Pty Ltd have signed a Memorandum of Understanding for developing collaboration between the two institutions resulting in this Software Testing course.

2 Overview of the Software Testing Course

Traditionally, the software engineering courses teach how to design and develop software application with minor emphasis on testing and maintenance. However, testing is an important activity both at the development and maintenance phases of software development. Industry data show that around 60% of software costs go into the maintenance phase of the software life cycle. Hence, it is important that software engineering students learn systematic testing of software systems [8].

Published research identifies numerous tools (AQTest, Test Mentor, JUnit, JProbe, MockObjects and many more) supporting requirements based testing. These tools provide support for test design, execution and defect identification. The tools provide little assistance to learn a particular testing method [5].

'Fundamentals of Software Testing' unit gives the students an excellent opportunity to get their hands dirty with real world testing problems. Whether student's desire is to be a career test analyst or simply wishes to expand the software engineering effectiveness, this unit provides real world, usable skills that are increasingly desirable to employers and the industry as a whole. This unit will enable students to develop a good understanding of software testing through both theoretical and practical application. Students will also learn the importance of exclusive Test Environments and how to develop a Traceability Matrix from Requirements through to Test Cases.

¹ <http://www.dfat.gov.au>

2.1 Learning Outcomes

This course allows students to experience the role of software tester in a software products lifecycle. The lecture session introduces them to the topic with practical examples of live systems and the lab session introduces the student to use and apply testing techniques in a structured manner. The goal is to prepare students for Software Testing for real time applications and hence they will need to demonstrate a broader understanding of importance and state-of-the art of software testing as practiced in the industry at the end of the course.

S.No	Goal	Theory Session	Lab Session	Outcome
1	Demonstrate their understanding of different types of testing	Topic 1	Lab Session 1	Goal achieved
2	Demonstrate their understanding of the entire life cycle of Testing	Topic 2 and 3	Lab Session 2	Goal achieved
3	Design and prepare Test Cases	Topic 5	Lab Session 4, Lab Test 1, Lab Test 2 and Mini Project	Goal achieved
4	Execute Test Cases and Capture Test Evidences	Topic 7	Lab Session 5, Lab Test 2, Mini Project	Goal achieved
5	Demonstrate their understanding of the defect life cycle, Severity, Managing and tracking defects to closure	Topic 6	Lab Session 6, Lab Test 2 and Mini Project	Goal achieved
6	Demonstrate their understanding of the importance of exclusive Test Environment for Testing	Topic 7	Lab Session 5 and 6, Lab Test 2 and Lab Test 3	Goal achieved
7	Create Traceability Matrix (from Requirements to Test Cases)	Topic 9	Lab Session 7, Lab Test 2 and Lab Test 3	Goal achieved

2.2 Lecture

Lecture sessions consisted of the following topics:

1. Types of Testing – covering Black box, Functional ,and Regression etc
2. SDLC – covering Waterfall Model, V-Model, Extended V-Model etc
3. Testing Life Cycle – covering Test Initiation, Test Preparation, Test Execution, Test Closure
4. Software Test Plan – Scope, Strategy, Schedule, Entry Criteria, Exit Criteria
5. Test Design – including Test Case writing, Test Data creation, Reviews

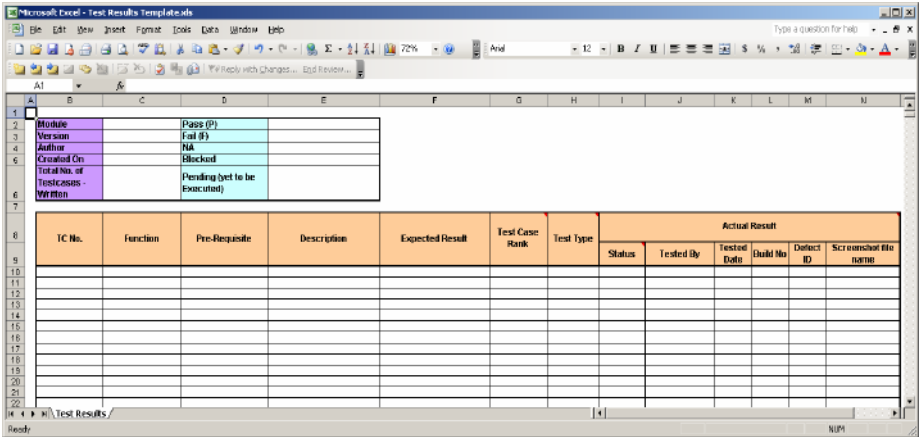


Fig. 1. Template to capture Test Results

S.NO.	Use Case Number	Use Case Name	Use Case section Number	Functionality Reference Number	Functionality	Test Case Numbers

Fig. 2. Template for Traceability Matrix

- 6. Defect Life Cycle – covering Defects creation, tracking to Closure
- 7. Test Execution – Importance of a separate Test Environment, covering executing the Test Cases, recording the Test results, capturing Test Evidences, Raising Defects on failure
- 8. Test Management – Covering all aspects of Test Management such as Risk Management, Schedule Management, Resource Management, Tracking the project for Efforts, Time, Cost and Quality
- 9. Traceability Matrix – Mapping the requirements to Test Cases and ensuring 100% Test Coverage
- 10. Test Closure
- 11. Introduction to Test Automation

Lecture notes were given in the form of detailed presentation with sample documents from real-life projects, enabling the students to understand the exact procedures, processes and templates. Deliverables such as Test Cases, Test Data, Test Plan, Defects Summary, Test Results were discussed through interactive sessions.

2.3 Lab Session

Lab sessions were conducted purely on the basis of making the students to have hands-on experience in testing the real time applications as follows:

1. ***Different Types of Testing*** – students were given different web applications to try out different types of testing and to raise defects if any. This made them understand different testing types
2. ***Demonstration of the Testing Life Cycle*** – students were taken through the testing life cycle by a demo of a real time application at the use cases, test cases corresponding to the use cases, test execution steps with results, test evidences and the mapping of the same to defects wherever the test cases failed
3. ***Preparation of Functionality Understanding document*** – students were given a module of a real application, were asked to understand the functionality and create documentation to articulate / describe the same in the form of functional flow with validation. This was useful to them to have learnt as to how application functionality understanding is core in the overall testing life cycle
4. ***Test Design*** – Students were made to create Test Cases covering all business-critical functionality, with necessary test data
 - 4.1. by reading through a Use Case document;
 - 4.2. they also learnt as to how to do the same by looking at an application instead of Use Case captured Test Evidences
5. ***Test Execution*** – A real time application was provided to the students for testing. They executed the Test Cases, recorded Test Results and Captured Test Evidences
6. ***Defect Management*** – A real time application was provided to the students for testing. They executed the Test Cases and raised defects whenever Actual Results observed did not match Expected results
7. ***Traceability Matrix*** – Students were asked to map the Test cases created in the previous sessions to the requirements

Since the Use cases and the application were from the University, students had to work on the Lab sessions on their own (these were not available either in the text book or in internet) making them learn the art of Test Design and Execution in a practical way.

2.4 Assessments

Since software testing is the process to assess the quality of software [9], we decided to assess the quality of learning by testing their skills as testers. We created assessments from the view of students going into the industry to work on real projects. The assessments were continuous assessment with no final exam to give a flavor of what to expect in real life projects.

Technical Report: The first assessment was a technical report where students had to do a survey on the state-of-the art of software testing as practiced in the industry. They had to pick organizations from Sydney and survey their software testing maturity.

Lab test 1 – A Use case document for a real application was given to the students using which, they had to create a Functional understanding document and Test Cases

Lab test 2 – The students had to demonstrate understanding of Software Testing Life cycle which includes writing Test cases, Creation of Test Data, Executing Test cases,

updating test results and creation of defects for a content management system created at UWS. The updating of test results included capturing screenshots of the defects raised as a part of the documentation. This application is being used currently to create applications at UWS. The templates of the documents to be created were provided to the students.

Mini Project – The major assessment was a project again given on a real application being developed at UWS for managing Staff Teaching Allocation. Student had to work in groups to perform complete Test Management of the application with no documentation provided. We called this assessment a Mini Project as it is a miniature of what they would face in real world of applications. Students had to create understanding document before design and execution of test cases and test data. The final deliverable was a report including the outcome of the test execution.

3 Challenges and Solution

Elbaum et al. identifies many challenges in incorporating testing into the curriculum [10]. However, we faced another challenge: making students understand the importance of testing. Generally they think that software testing is not rocket science and tend to give less importance to this. They were made to understand the importance by sharing the challenges faced in the industry due to injection of defects during various phases of SDLC, the cost of the same, software disasters etc. Studying a course on testing is relevant at this stage as it is likely to become an important part of Software's lifecycle as well as most graduates involved in software applications [10]. It became evident to the students that software testing is as much a professional discipline as development and can present interesting challenges and solutions after listening to the different stories of software testing in the class. They were made to realize that challenges for software testing becomes more and more due to more complex software systems being developed and also the expectation that software should run correctly even when the code is multithreaded, when the data is passed back and forth in real time across distributed systems, and when the software runs for longer and longer times without re-initialization.

4 Student Feedback

Student feedbacks have been very favorable to the offering of the course. Student reaction and participation in the lectures and labs confirmed the need for offering this course to software engineering students. They confirmed that they got an opportunity to learn concepts by practice in this course as compared to a theoretical learning. They thoroughly enjoyed the industry flavor to the course by the involvement of Polaris Pty Ltd. The outcome was successful to the extent that Polaris and few other organizations are going for recruitment of software testers from this group of students. The students gained not only the testing skills but employment as well from this unit.

5 Impact on Australian Markets

Australia depends on software in many areas starting driving cars to shopping on the internet etc, resulting in the users tolerance level decreasing on defective software. This necessitates that every software needs to be tested thoroughly prior to releasing it to production and hence there is a lot of importance given to effective and structured software testing[11]. Generally Software Testers were not available in Australian Markets to meet this growing demand. Even if there were a few available, most of them were looking at the tester's job as an entry point to get into a developer's position. The Software testing course at UWS has been able to make the students understand that the testers role in the software development life cycle spans across the entire life cycle and hence there is a lot of opportunity for them not only to learn the technology, but also the domain and the various phases, making it attractive to the students so much that most of them are keen to become software testers.

On completion of this course, the students are in a position to jump into industry and start working as software testers, since they have already undergone the testing life cycle in a practical way. This facilitates the software industry in Australia to be able to have a steady pool of software testing skills, thus, reducing the gap between the demand and supply for software testers in the Australian IT Industry.

References

1. Bentley, L., Whitten, J.: Systems Analysis and Design for the Global Enterprise, 7th edn., p. 747. McGraw Hill, New York (2007)
2. Cortex, I.: Statistics over IT projects failure rate (2004), http://www.it-cortex.com/Stat_Failure_Rate.htm
3. King, L.: UK government blows money on cancelled IT projects. CIO Business Technology Leadership CIO Business Technology Leadership CIO Magazine (2008), <http://www.cio.com.au/index.php/id;418199976>
4. Patton, R.: Software Testing, 2nd edn., p. 389. Sams Publishing (2006)
5. Collofello, J., Vehathiri, K.: An environment for training computer science students on software testing. In: Proceedings of 35th Annual Conference on Frontiers in Education, 2005. FIE 2005. (2005)
6. Gates, B.: Q&A: Bill Gates on Trustworthy Computing. In: Information Week (2002)
7. Duernberger, P.M.: Software testing applications in a computer science curriculum. In: Northcon/1996 (1996)
8. Ramakrishnan, S.: An Internet Environment for Learning Software Testing Processes. UniServe Science News 13 (1999)
9. Meyer, B.: Seven Principles of Software Testing. Computer 41(8), 99–101 (2008)
10. Elbaum, S., et al.: Bug Hunt: Making Early Software Testing Lessons Engaging and Affordable. In: 29th International Conference on Software Engineering, 2007. ICSE (2007)
11. Lingfeng, W., Tan, K.C.: Software testing for safety critical applications. Instrumentation & Measurement Magazine 8(2), 38–47 (2005)

Intelligent Recruitment Services System

Tetyana Shatovska, Victoriya Repka, and Iryna Kamenieva

Kharkov National University of Radioelectronics, Lenina av. 14, 61166, Kharkov, Ukraine
irina.kamenieva@gmail.com

Abstract. The Carrier Centre provides informational, analytical and organizational support of students and graduates' job placements. With this view the information system for supporting its main activities has been developed. Nowadays the system strengthens links between students and companies and serves as CVs and vacancies repository on the one hand. On the other hand in order to provide the effective decisions on employment the system should act as a virtual recruiter which takes into account students' personal abilities and preferences, available jobs, Company profiles, local labour market infrastructure, industrial and technological trends, account job specification, available human resources. This paper presents the intelligent management system for supporting recruitment services based on text mining methods.

Keywords: Clustering, text mining, similarity, documents.

1 Introduction

Nowadays a major social problem Ukraine is faced with is the one of formal unemployment among young people. Even holding University degree young professionals quite rarely will find the job which would be adequate to the obtained major. In particular, it is rather complicated to find an appropriate position on graduation. The problem could be solved by a closer co-operation between Universities and enterprises at the stage of Company's new high-tech positions strategic planning. Such a co-operation would be beneficial for the both sides and would allow students to get high-quality positions in private and public sectors. Two main classes of services provided by the Carrier Centre are to help the educated professionals to find appropriate jobs and to help companies to find right professionals to fill vacant posts.

The University Carrier Centre provides students with consulting and solution services taking into account their personal abilities and preferences, available jobs, current job specification, available human resources, applicants' profiles based on University degrees, University and Company profiles, National educational policy and standards, local labour market infrastructure, industrial and technological trends. Also the University Carrier Centre does a lot of labour market research like analysis of students' placement through practical bases, analysis of specialists and masters' employment at enterprises, weekly statistics of student's employment, analysis of tendencies in labour market demand for professions. This means that Carrier Centre makes analytical research and offers solutions for students' competitiveness growth

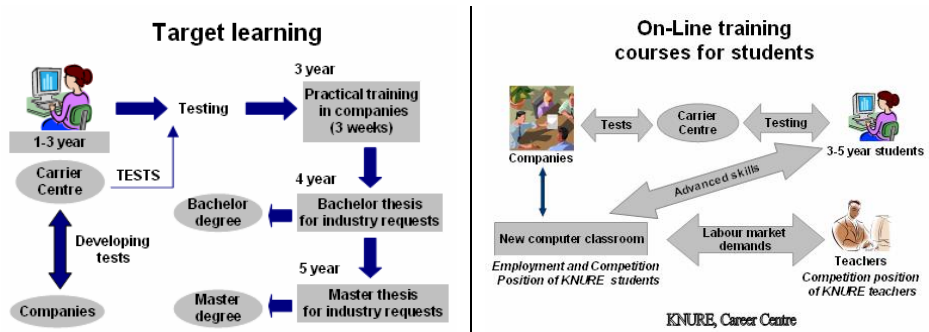


Fig. 1. Stages of competitiveness growth

(see Fig.1). Hardly would these activities be accomplished without an appropriate modern analytical information system.

One of the system modules: *employment virtual consultant*

The CV description is a challenge. Each of us is individual in delivering information. University Career Centre manager looking through and analyzing dozens of students' CVs every day has to face different writing and composition styles; format, fonts and logic structure of the CV's are absolutely arbitrary. Though companies may have their own structure of CVs some of them want to review and analyze the individual style and logics of CVs written by students as CV-writers with low experience. And here a system onto which managers' daily routine on CVs and company vacancies processing can be shifted would be of utmost help. The two main features of such system are applicants and company vacancies data automated gathering, CV's and vacancies clustering, and CVs - favourable vacancies automatic matching. In other words this system shall work as a virtual intelligent web-consultant for students. Though it should be noted that the criteria on which student CVs are selected by companies are always subjective.

Nowadays the process of comparing the new CVs with the previous ones as well as classifying CVs are totally manual jobs. The number of CVs grows (and the number of CVs written by the same applicant, too) while their processing deadline is reduced. As a result it is hardly possible to process the whole stream of incoming CVs manually. The creation of an intelligent web-system as of a consultant on recruitment will help to solve the following problems:

1. To analyse the structure and recognize the domain of CVs for their formalized representation.
2. To collect CVs and job presentations automatically from certain web-sites and add them to database.
3. To classify all the CVs and job presentations according to their subjects.
4. To eliminate duplicates.
5. To conduct flexible search according to user's inquiries.
6. To rank CVs and job presentations inside their groups taking into account the current hierarchy of a subject domain and using a matrix of skills and abilities.

7. To match job descriptions with the best available student CVs.
8. To annotate CVs and CV groups.

2 Overview

Now processing of applicants' information is partially carried out by hand that leads to information distortion. However automatic information extraction is not always correct either, therefore fully-automated mode will not solve this problem. The most suitable would be the automatic mode with manual acknowledgement. In the majority of cases automatic processing yields good results but in case when the system fails to distinguish correctly some parts of a CV the manager carries out its marking manually. Thus, the system receives another copy of training sample which will be used in the next training phase. Also the system should be able to check CV's consistency. For example, it is used to check the intersection of job periods in different places, to check the skills claimed in CVs and in real projects descriptions. As a basic function the system should attribute the incoming CVs and job descriptions to certain groups and to update the database.

There is no fixed pattern of CV-writing; therefore generally we consider that applicants and companies send their CVs and job descriptions in any form. Usually CVs consist of some parts or in other words structurized in what is referred to as logical blocks. It enables their (logical blocks) allocation for text clustering methods [1, 2, 3, 4]. In spite of the fact that CV writing styles may strongly vary general blocks can be extracted like Surname, Sex, Date of Birth, Previous Employment Description etc. Therefore we allocate common sets of attributes which can be found in the majority of CVs.

In order to provide module flexibility we do not use fixed templates either rules of data extraction from CVs and job descriptions. Instead the model of the CV and job description is created. On Fig. 2 it is shown the process of text information analysis. We use the approach of *adjusting* each CV to the constructed model of CV. For some final objects the set of rules for information extraction should be created. If some blocks have been distinguished incorrectly the module turns back to the training mode and creates the additional rule of information extraction which is put down to the knowledgebase. When some new CVs blocks emerge (e.g. the information on additional interests) in the mode of model editing the new elements are added and the extraction rule is constructed. Then all CVs are updated with the link to the new feature. This module can be adjusted to the extraction of any data from job placement domain – CVs, analyses of questionnaires etc.

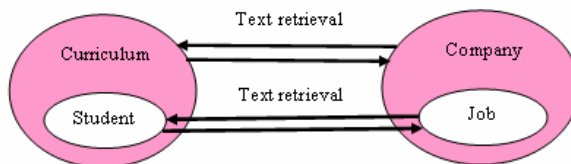


Fig. 2. Text information analysis

3 Method Description

3.1 Curriculum Vitae Clustering

Clustering is a process of grouping data into classes or clusters so that the objects within a cluster have high similarity in comparison to each other, but no or low similarity to the objects in other clusters. The list of classes is defined in advance and includes all the necessary education programmes in our University: Management, Mobile communication, Computer science, Radioelectronics etc. After the processing each CV is presented in the following scheme: a key-value, as $R = \{ri\}$, where ri – the CV, $ri = \{<key, value>\}$, where $i=1 \dots n$, n – number of attributes.

The description model is the same for all CVs. For each cluster the rule of CV frequency in a certain group was defined as $F = \{fi1, \dots, fin\}$.

Applying these conditions, we will receive a set of intersecting subsets $Ci \cap Cj \neq 0$, where $Ci = fi (ri)$. It is shown in Fig. 3.

For each attribute the measure $TF-IDF$ was applied [2,3]. Each CV or vacancy d is considered to be a weighted vector in the term-space and each document (vacancy or CV) can be presented as $tf1 * \log(n/df1), tf2 * \log(n/df2), \dots, tfm * \log(n/dfm)$, where tfi – is a frequency of i -th term in the document and dfi – is a number of documents which contains i -th term, and n – is a total number of documents in a sample. Each vector of the document should be normalised, $\|dfi\|_2 = 1$.



Fig. 3. Initial clustering

Let's take a popular measure of similarity for text (which normalizes the features by the covariance matrix) clustering is the cosine of the angle between two vectors.

Similarity measure

For similarity estimation the cosine similarity method is used, which is defined as in equation (1).

$$\cosine(d_i, d_j) = \frac{\langle d_i, d_j \rangle}{\|d_i\|_2 \times \|d_j\|_2} = \frac{\sum_{t=1}^t d_i \times d_j}{\sqrt{\sum_{t=1}^t d_i^2} \times \sqrt{\sum_{t=1}^t d_j^2}} \quad (1)$$

where d_i and d_j – components of vector documents, t – is a vector dimension.

Then the total $TF-IDF$ weight is calculated to classify the CV and job descriptions.

3.2 Clustering CVs Using Integrate Approach

Inside each cluster we define the conditions to divide CVs and job descriptions into subclusters to group a subcategory. These grouping conditions are defined by the user. In other words each CV or job description is an object with attributes where attributes are properties of the CV or job description, for example the description of certain skills. At the first stage of clustering we used a hierarchical approach [5,6]. It creates a hierarchical decomposition of the given dataset.

We integrate hierarchical agglomeration and iterative relocation first by using a hierarchical agglomerative algorithm with UPGMA method [6,7] and then refining the result using our iterative approach [8,9], similar to the Chameleon clustering [10]. At the final iteration of algorithm it determines the similarity between each pair of clusters by taking into account both their relative inter-connectivity and their relative closeness.

In our algorithm at the first phase we construct an asymmetric k-NN graph and there exists an edge between two points if for one of them there is the closest neighbour among all existing neighbours according to the value of k. Note that the weight of an edge connecting two objects in the k-NN graph is a similarity measure between them, as usually a simple distance measure (or inversely related to their distance).

We compute the weight of an edge as the weighted distance between objects. During coarsening phase the set of smaller hypergraphs is constructed. In the first stage of coarsening process we chose the set of vertices with maximum degrees and matched it with a random neighbour. In the other stages we visited each vertex in a random order and matched it with adjacent vertex via heaviest edge. Note that usually the weight of an edge connecting two nodes in a coarsened version of the graph is the number of edges in the original graph that connects the two sets of original nodes collapsed into the two coarse nodes. In our case we compute the weight of the hyperedge as the sum of the weights of all edges that collapse on each other during coarsening step. We stop the coarsening process at each level as soon as the number of multivertices of the resulting coarse hypergraph has been reduced by a constant less than two.

On the next level of algorithm we produce a set of small hypergraphs using k-way multilevel paradigm [11]. We start the process of partitioning by choosing most heavily multivertices k, where $k = 8, 16, 32$. After that we gathered one by one all the neighbours from each chosen vertex and obtain the initial partitioning w.r.t to the balancing constant. The problem of computing an optimal bisection of a hypergraph is NP-hard. One of the most commonly used objective functions is to minimize the hyperedge-cut of the partitioning; i.e., the total number of hyperedges that span multiple partitions [11]. In our experiments we use a greedy refinement algorithm developed by George Karypis [11], but as the gain function for each vertex we compute the differences between the sum of the weights of edges incidents on vertex that go to the other partition and the sum of edges weights that stay within the partition. We choose the vertex with maximum positive gain and move it if it results in a positive gain, so we work only with boundary vertices.

After the partitioning of hypergraph into a large number of smaller parts we start to merge the pairs of clusters for which both relative inter-connectivity and their relative closeness are high. In our research we use George Karypis formula to compute the similarity between sub-clusters and modified expression by changing the relative

inter-connectivity into a new expression that estimates the average weights of edges in each sub-graph and the number of edges that connect two partitions to the number of edges that stay within the smallest partition. Experimental results showed that this method is not sensitive to the value of k and doesn't need a specific k -nearest neighbour graph creating [7].

The CV is distinguished if the subcluster is defined and in appropriate way is saved into system. In Fig.4 the result of CVs clustering by using of above mentioned algorithm is shown. After clustering some clusters were taken such as: "System analysts" with no subclusters, "Engineers" with 3 subclusters (as you can see on Fig.4), "System Managers" with no subclusters and so on.

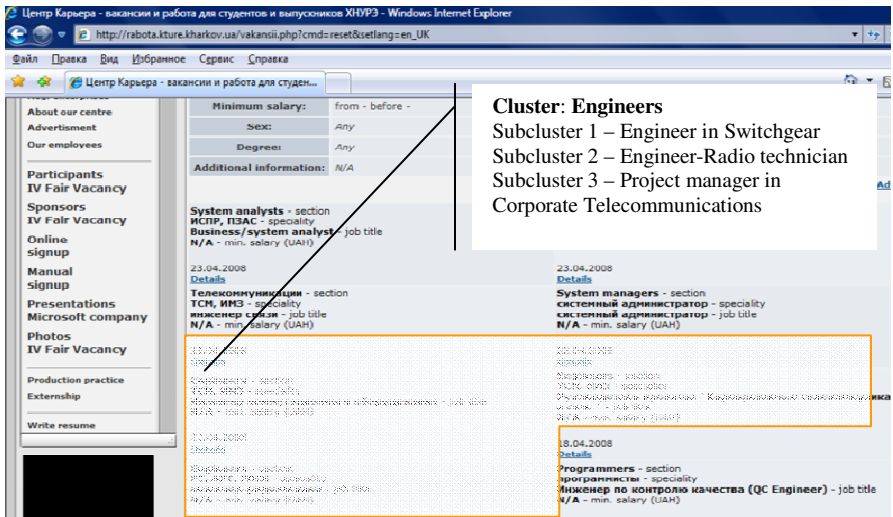


Fig. 4. CVs clustering

3.3 Applicants' CVs Annotation

Systems of texts processing use different approaches to text annotation. The most widespread way is the list of keywords. This way is simple for implementation, but there is lack of self-descriptiveness. Another way is automatic abstract construction. This way gives rather clear abstract, but is combined algorithmically. Considering a problem domain, it is offered (among other methods) to annotate applicants' CVs by adding a subset of the attributes from common blocks of all CVs like skills and abilities of candidates.

In our experiments 200 CVs were manually selected and marked. The marking included allocation of blocks "Education", "Experience" and "Other". Sections like «Contact information», "Hobby" etc was entered into "Other" block. In this part of our experiments we wanted to create a list of keywords for each block described above. At the initial step of pre-processing the unprintable symbols, stop words, marking symbols and also superfluous blanks, numbers and abbreviations were deleted. The second step was stemming. «Porter stemming» algorithm adapted for

Russian language was used [12]. It is also simple for implementation as it is constructed on heuristic rules of words truncation and does not require dictionary support. Unfortunately, in atypical words it makes errors, but it occurs seldom and does not influence the final result. After normalization the word is located in the list of keywords for the given block. For the generated keywords it is calculated as well the frequency of their occurrence. Sometimes it is mistaken as the length of lines shall be nothing more than three. As these lines do not concern semantic carriers of the block, it is possible to remove them. As the result of the second step we receive the list of bases keywords with frequency of its occurrence in the block text.

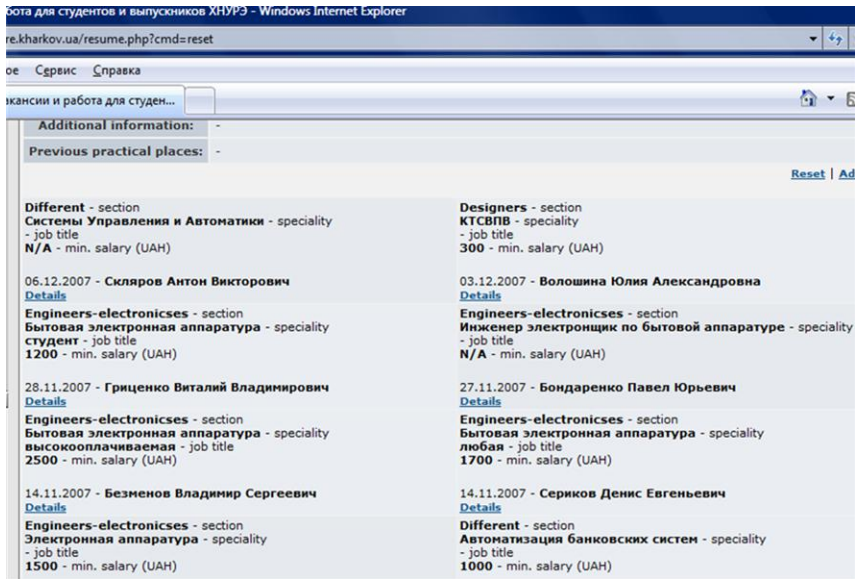
After the list had been formed it turned out that one word can belong to several lists. In this case it is not the unique characteristic of the block. To preserve uniqueness it was necessary to get rid of keywords intersection. Frequencies of such words were compared. The word remains in the group where the frequency of its occurrence is the highest and is omitted from the group where its occurrence frequency is lower. Thus we have received not intersecting sets of keywords with frequency characteristics for each group.

As practical experiments have shown, usage of the roots of keywords only does not give us exact splitting. Therefore to define broader blocks the phrases from blocks headings were used. At the stage of entrance files manual parsing the headings were separately allocated. Simultaneously for each block the list of keywords and headings was formed. The accuracy of blocks division using only keywords of headings was 80 %. If the heading border has not been distinguished the information was saved in system. Performing analysis we defined the possible splitting of CV's text into blocks. To confirm or correct this splitting the statistical approach on the basis of the available information on keywords was used. As a result we received the text broken into blocks Educations, Experience and Other. Splitting of the text into blocks occurred as follows. For each word we find its normal form using the aforementioned stemming algorithm [12]. Further we search the obtained normal form in the lists of keywords and if we find it we paint this word in the group colour.

In Computer Science the fact is the individual value of the data created or used by business process. The facts of our problem domain are: the date of birth, the marital status, University entrance and graduation dates, professional skills, citizenship, languages etc. Allocation of the facts occurs by the certain rules, constructed on the basis of regular expressions [13]. They are formed in the training phase.

Using CV as a common model makes it possible to reach the high quality of classification. The offered approach can be applied for analysing CVs and job descriptions and solving the problem of their comparison with those existing in the system. The method is used for compiling Top CVs List (the List of CVs which is in great demand among the employers).

The pilot version of the system was developed. On Fig.5 you can see the result of system work for compiling Top CVs list. The efficiency analysis of 500 CVs showed that in 87% of the cases splitting into blocks was performed correctly. And in 82% of the cases the facts were correctly allocated. The analysis of cases when the system could not break the text into blocks correctly was carried out: atypical styles of CVs, HTML-tables.



Additional information:	
Previous practical places:	
Different - section Системы Управления и Автоматики - specialty - job title N/A - min. salary (UAH)	Designers - section КТСВНБ - specialty - job title 300 - min. salary (UAH)
06.12.2007 - Скляров Антон Викторович Details Engineers-electronicses - section Бытовая электронная аппаратура - specialty студент - job title 1200 - min. salary (UAH)	03.12.2007 - Волошина Юлия Александровна Details Engineers-electronicses - section Инженер электронщик по бытовой аппаратуре - specialty - job title N/A - min. salary (UAH)
28.11.2007 - Гриценко Виталий Владимирович Details Engineers-electronicses - section Бытовая электронная аппаратура - specialty высокооплачиваемая - job title 2500 - min. salary (UAH)	27.11.2007 - Бондаренко Павел Юрьевич Details Engineers-electronicses - section Бытовая электронная аппаратура - specialty любая - job title 1700 - min. salary (UAH)
14.11.2007 - Безменов Владимир Сергеевич Details Engineers-electronicses - section Электронная аппаратура - specialty - job title 1500 - min. salary (UAH)	14.11.2007 - Сериков Денис Евгеньевич Details Different - section Автоматизация банковских систем - specialty - job title 1000 - min. salary (UAH)

Fig. 5. Top CVs

4 Conclusion

The article recounts the idea of developing an intelligent system to supporting student employment process. As a virtual consultant for employment it can facilitate the process of finding a job.

The offered approach can be used for the solution of classification problems, segmentation and allocation of the facts in other areas connected with documents circulation in recruitment services.

Universal model of CV and job description makes it possible to attain high quality of classification. The offered method helps to solve such tasks as applicant's CV annotation and automatic matching of CV with available vacancies. Integrated clustering approach for CV's similarity estimation was offered. On the basis of it the List of Top Actual CVs was formed.

References

1. Liu, B., Lee, W.S., Yu, P., Li, X.: Partially supervised classification of text documents. In: Salton, G., McGill, M. (eds.) ICML 2002. Introduction to Modern Information Retrieval. McGraw-Hill, New York (2002)
2. Yang, Y., Pedersen, J.P.: A comparative study on feature selection in text categorization. In: ICML 1997 (1997)
3. McCallum, A., Rosenfeld, R., Mitchell, T., Ng, A.: Improving text clasification by shrinkage in a hierarchy of classes. In: Proceedings of the International Conference on Machine Learning (ICML), pp. 359–367 (1998)

4. Toutanova, K., Chen, F., Papat, K., Hofmann, T.: Text classification in a hierarchical mixture model for small training sets. In: Proceedings of the Tenth International ACM Conference on Information and Knowledge Management, CIKM (2001)
5. Joachims, T.: A probabilistic analysis of the rocchio algorithm with TFIDF for text categorization. In: Proc. Of the ICML 1997, pp. 143–151 (1997)
6. Zhao, Y., Karypis, G.: Evaluation of hierarchical clustering algorithms for document datasets. In: Proceedings of the International Conference on Information and Knowledge Management (2002)
7. Zhao, Y., Karypis, G.: Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning* 55(3) (2004)
8. Shatovska, T., Safonova, T., Tarasov, I.: A Modified Multilevel Approach to the Dynamic Hierarchical Clustering for Complex types of Shapes. *Lecture Notes in Informatics (LNI) – Proceeding*, vol. P-107, pp. 176–186 (2007)
9. Shatovska, T., Safonova, T., Tarasov, I.: The New Software Package for Dynamic Hierarchical Clustering for Circles Types of Shapes. In: Proceedings of XIII International Conference KDS 2007, Varna, Bulgaria, June, pp. 125–129 (2007)
10. Karipys, G., Han, E.H., Kumar, V.: CHAMELEON: A hierarchical clustering algorithm using dynamic modelling. *IEEE Computer: Special Issue on Data Analysis and Mining* 32(8), 68–75 (1999)
11. Karypis, G., Kumar, V.: Multilevel k-way hypergraph partitioning. In: Proceedings of the Design and Automation Conference (1999)
12. Russian stemming algorithm (2005),
<http://snowball.tartarus.org/algorithms/russian/stemmer.html>
13. Keleberda, I., Repka, V., Biletskiy, Y.: Building learner's ontologies to assist personalized search of learning objects. In: ICEC 2006, pp. 569–573 (2006)

Challenges and Opportunities Relating to RFID Implementation in the Healthcare System

Belal Chowdhury and Clare D'Souza

La Trobe University, Melbourne 3086, Australia
{M.Chowdhury, C.D'Souza}@latrobe.edu.au

Abstract. In a health care context, the use of RFID (Radio Frequency Identification) technology can be employed for not only bringing down health care costs but also facilitating the automating and streamlining of healthcare management process (e.g., patient identification) automation in hospitals. The technology also has the potential to drastically reduce the occurrence of medical errors, decrease the time needed to locate precious resources, improve asset management and enhance patient safety. Despite the potential benefits, there are some significant challenges and obstacles with the deployment of a RFID-based system in the healthcare. The paper outlines the major RFID issues faced by healthcare providers. It addresses a case study on Hospital Information System (HIS) applications and examines the issues of RFID implementation in the current healthcare system.

Keywords: RFID, HIS, HMS andx Barcodes.

1 Introduction

The healthcare industry lags behind automating and streamlining the health management process despite the fact that many countries in the world allocate resources in this sector. Nonetheless, there is a shortfall of efficient and operational resources throughout this sector. While offering the finest benefits to some of its citizens, countries are burdened with skyrocketing health costs, rocketing insurance premiums, declining coverage and uneven access to quality care. Recent online survey conducted by the Health Care in America (2008) reveals that one-third of respondents report skipping medical care because of cost, and a quarter had serious problems paying for the care they needed. Overall 95% respondents say that they are somewhat or very concerned about being able to afford health insurance in the coming years and America's health care system needs fundamental change or to be completely rebuilt [1]. Besides rising costs, to add to the dilemma there are also errors that can be compounded. For instance, the Commonwealth Fund's 2007 International Health Policy Survey shows that 32% of U.S. patients with two or more chronic conditions reported a medical, medication, or lab test error in the past two years, compared with 28% of patients in Canada, 26% in Australia and 24% in UK [4].

Developed countries like Australia and others are currently facing a middle- and older-aged marketplace from a predominantly youth-driven marketplace. According

to the 'Year Book Australia 2008' published by Australian Bureau of Statistics recently, about 26% of the population is projected to be aged 65 years and over by 2050, in comparison to 15% of the population being aged 0-14 years. The effect will be particularly noticeable after 2011, when the first of the baby boomers reaches the retirement age of 65 [5]. As the aging population and people with chronic diseases survive longer, their health issues are likely to impose greater stress on the health system in Australia. Despite advances in medicine, chronic conditions continue to be major contributors to the burden of disease worldwide.

Medical experts predict that many healthcare systems will become unsustainable in the world by 2015 [2]. Although, people with chronic diseases need ongoing medical care, including monitoring, treatment, and coordinating among multiple healthcare providers (e.g., hospitals) that is likely to last longer period (i.e., more than one year) [3]. This indicates the dire requirement for long-term care for the aging population, efficient facilities and more resources. Mistakes (such as wrong medication) in medical care can occur anywhere in the health care system at hospitals, doctor's offices, nursing homes, pharmacies, or patients' homes. The medication error has been estimated to result in at least 80,000 hospital admissions and costs of at least \$350 million per year in Australia [6]. Reduction in error rates and efficiency in facilitating systems are key components of any managerial system. Change is possible by advances in technology, but it is being driven by market forces and societal desire to improve the health of a nation's citizens, while reducing healthcare costs.

There is also growing concern about maintaining one's health as population ages. Mobile/sensor technology is expected to provide real-time information about vital signs and other physiological indicators of one's health and fitness. Such monitoring systems are expected to find greater use in such applications in health facilities (e.g., hospitals, nursing homes, special accommodation facilities and rehabilitation hospitals). The paper recommends that the application of real-time monitoring system can be facilitated by the use of the mobile technology such as Radio Frequency Identification (RFID). RFID is one of the emerging technologies that elegantly provide a solution that can seamlessly integrate the captured data at various levels in the healthcare business processes with the backend databases, backend applications and decision support system [16].

As suggested by Chowdhury and Khosla (2007) that today's advanced technology is capable of uniting patient tags (RFID) and data processing into a single integrated system. The payoff can be dramatic, from saving countless hours of search time to eliminating hundreds of thousands of dollars in equipment replacement or excess rentals. Using RFID technology, a medical staff (e.g., a nurse) could monitor the status of an entire ward of patients more effectively and accurately. Outpatients could be monitored remotely, receiving nearly the same level of attention as those within the hospital system. That means RFID has the ability to deliver higher quality, at a more efficient cost, an issue that is of concern to governments worldwide.

Due to the larger amounts of data storage and capacity for interactive communication, RFID technologies are far more powerful than the conventional identification techniques such as barcodes. Unlike barcodes, RFID technologies do not need line of sight and the tag (RFID) can be read without actually seeing it [7]. The RFID tag's read rate is much faster than the barcode system. Some advanced RFID readers that can read up to 60 different RFID tags at approximately the same time, while a barcode

reader can scan only one item at a time [8]. Despite these potential benefits and clear advantages over bar coding, there are challenges or issues with implementing RFID-enabled system applications in the healthcare. The paper attempts to address and examine some of the important issues relating to RFID implementation in healthcare management systems (HMS).

The paper is structured as follows: section 2 illustrates challenges relating to RFID implementation in the healthcare system. In section 3, a case study of healthcare system is used to demonstrate the application of the practicality relating to RFID implementation issues. Section 4 provides a broad discussion of the results and section 5 concludes the paper.

2 RFID Implementation Challenges in HMS

There are key issues that present a host of challenges for the successful implementation of RFID technology in the HMS system. Some of the challenges relating to RFID implementation in HMS systems are as follows:

- a) *Privacy* – Privacy issues loom as one of the biggest concerns to the success of RFID implementation in a hospital system. Concerns have been raised about the right to privacy being compromised by the emerging technology. An intruder with unauthorized readers can intercept the communication between the patient tags and RFID readers, between readers and the back-end database system in hospitals, and can access sensitive patient information (such as patient ID, name, drug allergies, drugs that the patient is on today, blood group, and so on). Privacy advocates express concerns that placing RFID tags in common items (e.g., pharmaceuticals) may allow them to continue to be tracked once purchased by patients. A serious concern for patients is that once they have purchased items (e.g., sleeping pills from a hospital pharmacy), they do not want themselves or the purchased items to be tracked after passing the checkout [9].
- b) *Security of communication channel* - Most of the security threats in a hospital are attributed to the security of the communication channel between authentic RFID readers, and the patient tags through the air interface (i.e., wireless communication). A RFID patient tag reading occurs when a reader generates a radio frequency “interrogation” signal that communicates with the tag (e.g., a tagged patient), triggering a response from the tag [11]. Unauthorized readers can trigger this response by revealing the patient information such as the patient ID and it is subject to misuse by hackers and criminals. Further, with respect to Read/Write (reprogrammable) tags, unauthorized alteration of patient data is possible in the hospital information system.
- c) *Patient safety concerns* – improving patient safety is one of the nation's most pressing health care issues today. The concern of correct traceability and localization of the patients in RFID-based healthcare system. There are factors such as medical errors and drug-related injuries that not only add unnecessary cost to the healthcare system; they also affect overall public health. In addition, communication system between health professionals compromising patient safety.
- d) *Implementation Cost* – A recent survey shows that high cost remains the primary roadblock to greater RFID implementation in health care [10]. The cost of hospital

infrastructure (RFID hardware - tags, and readers, IT system, network infrastructure, system integration, process redesign, organization change, labor cost, and so on) is high. The cost of RFID-tags is far more than a barcodes system. Even though the cost of tagging is decreasing, it is still significant.

- e) *Data security issues* - Data security is a major issue for wireless due to the nature of the transmission mechanism (electromagnetic signals passing through the air) in a RFID-enabled healthcare system. The security of the patient database repository from unauthorized users (e.g. hackers, and others) is a major issue in RFID-based hospital systems. The transmission of the collected patient data from a RFID reader over an intranet/internet to a remote database is vulnerable to the interception, alteration or destruction of signals [13].
- f) *Lack of Standards* - Lack of standards is a major obstacle for the deployment and support for widespread use of RFID system in hospitals. Currently there is no consistent or common standard for the air interface for healthcare industry. Item-level tagging is also necessary for most of the hospital equipment or asset management processes where the payoff occurs. Without clear RFID standards and data ownership policies, investment of RFID system in healthcare has been a difficult task.
- g) *Patient Tag frequency and Serialization* - Frequency and serialization is also a significant issue in RFID-enabled healthcare system. Healthcare providers such as hospitals are very concerned as to which tag frequencies to use and where. With the serialization issue, they are concerned about what to include in a patient tag's serial number. Some want the tag serial number to contain intelligence (e.g., patient ID) information; others do not want the tag intelligence information, rather a random serial number.
- h) *Radio Frequency health Issues* - Concerns about PDA and cell phone based RFID reader can cause potential impact on a variety of radiation-emitting medical instruments or devices such as MRI units, x-ray machines, and pacemakers in a critical hospital environment. Many hospitals ban such phones in their hospital settings. Again, this has led some concern whether RFID devices can be used in proximity to medical equipment [19].
- i) Another area of interference is unrelated wireless networks. Communication between tags and readers are inherently susceptible to the radio frequency interference, especially when other systems are using the same frequency range within the hospitals.
- j) *Tags Read Rate* - RFID readers are not always able to read tags (e.g., patients, medical equipments or assets) on a 100% basis [10]; this becomes one of the major obstacles to RFID deployment in healthcare (e.g., hospitals) sector. Local electromagnetic interference (EMI) is one of the features, which affect tags' accuracy [12].
- k) *Presence of Metal Objects and/or Liquid Containing Items* - Healthcare management system is an area of operations that normally teeming with metal, liquid and harsh environments. Interference from metal objects (e.g., operating instruments) that generate electromagnetic energy [12], disrupts the RFID signal and liquids absorb radio frequency signals make it challenging for many healthcare providers (e.g., hospitals) to tag and track with their RFID-enabled HIS. The RFID tag is also affected by objects surrounding it especially metallic objects [11].

- l) *Lack of Trust and confidence* – Trust is another important issue influencing the internal health organization relationship with external health organization partnership. When the degree of integration between business partners is increased, partners could access patient's data without authorization that was inaccessible previously. Many healthcare providers (e.g., hospitals) are also reluctant to invest in a RFID technology not yet widely adopted.
- m) *Healthcare IT infrastructure* - Modifying existing business processes is a daunting task for healthcare providers in a hospital that usually entails changes in RFID technology investment strategies. RFID-enabled system implementations are also part of hospital IT infrastructure, which often necessitates significant work process. The lack of various components such as RFID tags, networked readers, RFID-based healthcare system applications, intranets (LANs and WANs) and back-end database servers may affect enhancements to facilitate the implementation process.
- n) *Job Loss* – The penetration of RFID into the hospital system eliminates or transfer jobs, it may have to be submitted to hospital staff negotiation. Many healthcare providers predict that the use of RFID in the hospital system may affect jobs due to the nature of the systems automation.

3 RFID in the Hospital Information System – A Case Study

To address the challenges and issues mentioned in the section 2, we present a case study on a Hospital Information System (HIS). We outline a RFID model for designing HIS, which can help healthcare providers (e.g., hospitals) to overcome these challenges by providing accurate, real-time information on patients as they move to the value chain and by automating related business processes. An application of the architecture is also described in the area of RFID-based HIS.

3.1 RFID Model for Healthcare Systems

A wristband (attached with RFID tag) can be issued to every patient at registration, and then it can be used to identify patients during the entire hospitalization period. It can also be used to store patient important data (such as patient ID, name) in order to dynamically inform health professionals before critical. RFID encoded wristband data can be read through bed linens, while patients are sleeping without disturbing them [15].

The main components of RFID-based HIS is shown in Figure 1. It mainly consists of patient tags (i.e., tiny chips) or RFID-enabled wristband, a reader (fixed and hand-held) and health care provider IT systems (i.e., Real-Time RFID-Based HIS). Each unique patient tag can be passive, semi-passive or active. Passive patient tags can be used for both reading/writing capabilities by the reader and do not need internal power (i.e., battery). They get energized by the reader through radio waves and have a read range from 10mm to almost 10 meters. We suggest the use of passive patient tags (13.56 MHz ISO 15693 tag) with the read range of one meter, and PDA/Next G Smart Phone RFID readers for the real-time Healthcare Management System (i.e., HIS) application [7].

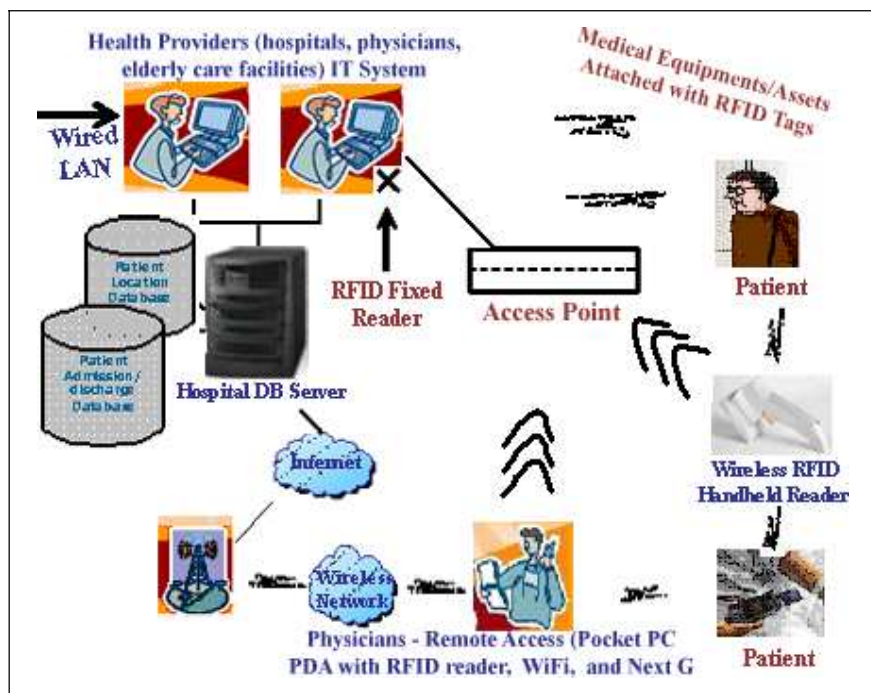


Fig. 1. Main components of RFID enabled healthcare system (e.g., HIS)

The passive patient tag (used in HIS with a frequency of 13.56MHz) antenna picks up radio-waves or electromagnetic energy beamed at it from an RFID reader device and enables the chip to transmit patient's unique ID and other information to the reader device, allowing the patient to be remotely identified. The reader converts the radio waves reflected back from the patient tag (i.e., wristband) into digital information [7] then pass onto HIS system for processing. Patient's basic information is stored in the back-end server for processing data. The patient database can also be linked through Internet wirelessly into other health centers databases [17] for retrieving patients past history. In addition, and with a whole real-time RFID-based HIS system approach and integration with other health care applications, there is the exciting potential of delivering more efficient patient care through higher levels of compliance with care pathways.

3.2 RFID-Enabled HIS Application

As the healthcare industry faces data integration issues, the RFID device management is a challenge while deploying RFID devices in their health provider's system. Multi-layer RFID architecture establishes an infrastructure to address such a challenge, to automate and simplify the functionality for building RFID-based solutions in the healthcare system. These integration layers (i.e., five layers) are namely, physical device layer, middleware layer, IT infrastructure management layer, data layer and graphical user interface layers.

The physical device layer consists of the actual RFID hardware components (such as RFID tag, and reader) that integrate with HIS for capturing data automatically. The middleware layer or framework is viewed as the central nervous system from the healthcare system perspective. It acts as the standard mechanism to get a quick connectivity between patient tags and RFID-enabled healthcare providers IT systems as shown in RFID model in Figure 1. The IT infrastructure management layer is responsible for managing and controlling the healthcare provider's IT components such as computers, back-end servers, networks, and printers. In addition, this layer enables data mapping, formatting, business rule execution and the service interactions with back-end databases. The data layer composed of a RDBMS (Relational Database Management System) and it interacts with a back-end database (i.e., Microsoft SQL server 2005 database) and includes a data query/loading approach using SQL (structured query Language) that supports high volumes of RFID data into a custom designed RFID database schema. Finally, the graphical user interface (GUI) layer is comprised of an extensible GUI (graphical user interface), which monitors patients or assets data. The GUI also helps in managing the information, generating various reports and analyzes the information at various stages in the entire value chain.

Figure 2 and 3 shows the RFID-enabled HIS application, which can be integrated with the healthcare providers IT System for capturing patients, medical equipment, or assets data automatically and wirelessly. The system is developed in Microsoft Visual Studio.net 2003 environment using Visual C++ (MFC). The RFID-based HIS application issues a unique tag ID to every patient with a wristband at registration/admission in hospitals is shown in Figure 2.

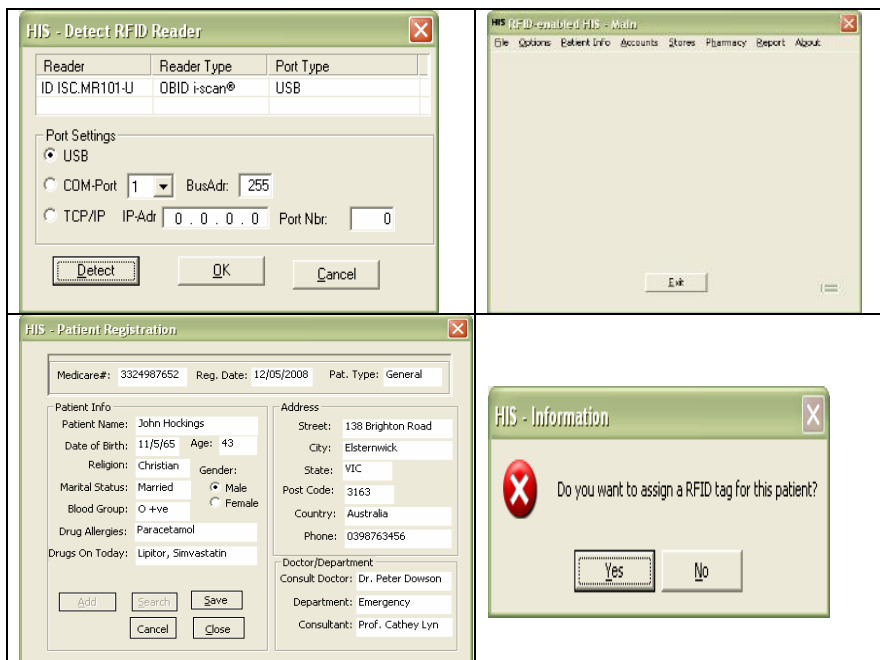


Fig. 2. RFID Patient Tag Assignment using HIS

The RFID then uses the tag ID as a key to information and perhaps other information (e.g. name, DOB, drug allergies, blood group, etc.) stored in the health providers back-end databases. The wristband is used to identify patients all the way from observation, investigation, and treatment to discharge while in hospital. For example, an RFID patient tag only contains a unique tag ID, which a HIS application uses to retrieve a patient record stored in the database. When a patient appears with a wristband within a reader (i.e., placed in hospital wards, clinical labs, pharmacies, radiology departments, and so on) read range, the application reads and lists the tagged patient IDs, names and displays the patient admission information automatically on a selection of a particular patient as shown in Figure 3. The RFID patient tag (wristband) can be removed from the patient on discharge from the hospital is also shown in Figure 3. The available wristband from such a patient can be reused.

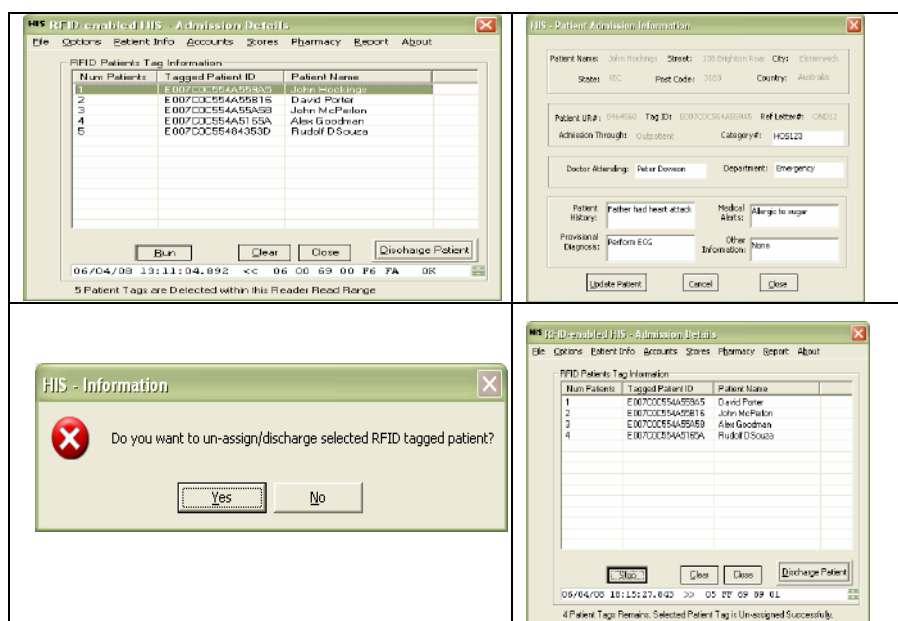


Fig. 3. Un-assigned a RFID Tag on a Patient Discharge

4 Discussions and Results

Although in section 2 a series of issues were addressed, it is beyond the scope of this paper to examine all of them as the intention of this case study is to demonstrate how to apply RFID implementation techniques in the HIS system. We have highlighted some of the most significant issues such as privacy and security, patient safety, RFID tag read rate and standard. Further research will be required to address the remaining issues.

- a) *Privacy and Data Security* - In a RFID enabled HIS, communication between RFID tags and readers, readers and back-ends database systems are one way. Our

tags are passive, inexpensive and have a minimum amount of memory. We are keeping very little information in the RFID patient tag e.g., tag ID only and any private information is kept in a separate secure back-end database system to protect an individual's privacy. In our HIS, even if RFID tags hold patient data such as patient ID, name, drug allergies, drugs that the patient is on today, and blood group, the reader must be authenticated with a secret reference number (e.g., reader serial number) by the tagged items before reading them. Data in a patient tag is encrypted so that only authorized user (e.g., health professionals) can obtain the original patient data. For the secure transfer of patient data from RFID-enabled HIS to back-end database server, we are using a Hash Function-based Mutual Authentication Scheme [14]. This scheme, utilizing a hash function, is widely used for secure communication between readers and back-end SQL server databases in a RFID-based HIS.

In the worst-case situation, if an intruder intercepts and gets the patient ID, he/she gains nothing because the tag does not contain any additional information. The intrusion of personal privacy and security issues can be overcome through open and transparent communications.

- b) *Patient Safety* - Medical errors in hospitals have become a leading cause of death, killing more people each year than AIDS in the world. These medical errors can be classified as poor-decision making, poor communication, inadequate patient monitoring, patient misidentification, inability to respond rapidly and poor patient tracking [18]. RFID-based HIS provides the opportunity to track and monitor tagged patients in real-time which allows attending health professionals (e.g. doctors, physicians, nurses) to monitor patients data and to guide them in a very efficient manner. That obviously reduces a range of medical errors (such as in prescribing, accessing incorrect patient records, operating on the wrong patient, or locating a wrong medical equipment or asset) to improve patient safety by locating the patient and/or the appropriate medical staff more quickly and also improve the communication between health professionals in the hospital system.
- c) *RFID Tag Read Rate* - One of the primary reasons affecting the RFID tag read rate is poor positioning (i.e., tag misalignment), which results in a lack of sufficient energy. The other factors that affect an RFID tag read rate includes collisions on the air interface, tag frequencies, tag detuning, reader capabilities, operating environments (i.e., interference from other devices, temperature, humidity, and vibration), metal object and liquid containing items in the vicinity of the RFID-based HIS. It has been observed that the presence of metal and/water in the RFID tag vicinity causes a failed tag read within healthcare business processes. As radio waves bounce off metal and are absorbed by water at UHF (ultra-high frequencies), it distorts the magnetic flux, thus weakening the energy coupling to the tag [12]. This makes metal medical equipments difficult to tag and track with a RFID-based HIS system.

In our RFID-enabled HIS, we have placed non-conductive material between the tag and the metal object to avoid scanning problems and improve tag reading rate. This technique is only useful if the size (e.g., one square feet) of metal object is small. In the case of multiple patient tags reading the reader detects a collision and performs an anti-collision to address collision issues.

d) *Standard* - The standard in a RFID system is critical for promoting healthcare industries in their value chain development. RFID Standards are more likely to provide health industries with complete visibility of their value chains, which often stretch across countries, industries and companies. There are number of well-established open standards such as the International Organization for Standardization (ISO) and EPCGlobal. As our RFID tags are ISO standard in HIS, the reader support and communicate with these (ISO 15693) tags effectively. Our proposed system uses high frequency (i.e., 13.56 MHz) passive tags, which have a unique serial number with the read range of one meter. The ISO enables more salient and detailed information to be captured for a patient or medical equipment. The tag can then be programmed to hold important patient information such as a patient's ID, thus enabling greater person or equipment accountability and safety. RFID-based HIS was developed on IEEE 802.15.4 standard, where a reader retrieves different tags (patient, medical equipment, assets). In addition, regulatory forces (such as Department of Health) can impact the quality of patient care and services in light of the common standards required for compliance (such as HIPAA - Health Insurance Portability and Accountability Act – 1996, JCAHO - Joint Commission on Accreditation of Healthcare Organizations-1951 in USA) and accreditation to ensure interoperability and assist healthcare providers to build the technology, and the security to enforce them.

As the RFID technology becomes more widespread, the unit cost of tags and readers will continue to fall. As the RFID tag and reader prices decline and applications become more robust, RFID technology will be increasingly adopted. Even though RFID tag operating frequencies and serialization differs between the USA, Europe and other regions, support for open standards such as ISO and multiple frequency readers make sense for global healthcare systems.

Healthcare industries need to make changes in their business processes to realize the full benefits of RFID technology. For example, tagging patients or medical equipments and placing RFID readers in doorways at hospitals will require an initial deployment and a change in protocols for locating assets. Although, it will be critical to be able to understand the potential risks and limitations of the technology in order to improve reliability and to assure healthcare providers and encourage them to place their confidence in the technology. These risks are easy to manage once they are understood. While the initial training requires coordination and effort, the decrease in time spent looking for assets and increased asset utilization improves the overall efficiency of the hospital system. Experts predict that RFID will be a major advance in healthcare management system, but healthcare providers need to do considerable upfront planning and testing in implementing and/or integrating RFID technology successfully.

5 Conclusions and Future Work

In this paper, we have outlined challenges relating to RFID implementation within healthcare system. A descriptive case study in the area of healthcare system (i.e., HIS) has illustrated the RFID implications for RFID. We have proposed a RFID model for designing a HIS to help healthcare providers (e.g., hospitals) to meet/overcome

challenges by providing accurate, automatic and real-time information on patients, medical equipments or assets as they move to the value chain. An application of the architecture is also described in the area of RFID-based HIS.

RFID-based healthcare system can provide facilities to improve overall safety and operational efficiency because it operates through air interface (i.e., wireless communication) while providing read/write capabilities for dynamic patients, medical equipments or assets tracking. To do that healthcare decision makers (e.g., Medical and health services managers) can move to developing RFID-enabled applications and integrating RFID data into existing applications to drastically reduce the medical errors, to reduce the time needed to locate precious resources and to dramatically improve asset management.

The coordination of healthcare providers seems to be a major factor for influencing the speed and ease of the RFID introduction process. As the sharing of open RFID system development and crucial information along the healthcare system requires common standard health procedures and regulations, trust from the health staff and the patient need to be in place. Despite the efforts of the two large standardisation bodies ISO and EPCGlobal, differences in regional standards remain a hindering factor in global RFID healthcare system applications. Therefore, further research in RFID tag standards, economic, social, organisational and other issues, relating to the coordination and integration of healthcare providers with regards to RFID applications and deployment should be conducted.

References

1. Antwerp, G.V.: Patient Centric Healthcare, March 25 (2008) (accessed April 10, 2008), <http://patientadvocate.wordpress.com/2008/03/25/poll-shows-real-issues/>
2. IBM Healthcare and Life Sciences specialist, Patient-centric: the 21st century prescription for healthcare, 1133 Westchester Avenue, White Plains New York 10604, U.S.A (2006)
3. Gross, P.F., Leeder, S.R., Lewis, M.J.: Australia confronts the challenge of chronic disease. *The Medical Journal of Australia (MJA)* 179(5), 233–234 (2003)
4. Davis, K.: Health Care: Solutions Without Borders, The Commonwealth Fund's 2007 International Health Policy, November 26 (2007) (accessed April 11, 2008), http://www.commonwealthfund.org/aboutus/aboutus_show.htm?doc_id=597055
5. ABS (Australian Bureau of Statistics), Year Book Australia, 2008, February 7 (2008) (accessed April 11, 2008), <http://www.abs.gov.au/ausstats/abs.nsf/mf/1301.0>
6. Bruce, H.B.: Safety and quality in Australian healthcare: making progress. *The Medical Journal of Australia (MJA)* 174, 616–617 (2001)
7. Chowdhury, B., Khosla, R.: RFID-based Real-time Hospital Patient Management System. In: *Proceedings of the 6th IEEE/ACIS International Conference on Computer and Information Science, and International Workshop on e-Activity 2007*, July 11–13, 2007, pp. 363–368. IEEE Computer Society, Los Alamitos (2007)
8. Rao, S.: Supply Chain Management: Strengthening the Weakest Link!, Team Leader for Industrial Automation, March 1 (2004) (accessed June 01, 2007), <http://hosteddocs.ittoolbox.com/SR032604.pdf>

9. Michael, K., McCathie, L.: The pros and cons of RFID in supply chain management. In: Proceedings of the International Conference on Mobile Business (ICMB 2005), July 11-13, 2005, pp. 623–629. Copyright IEEE (2005) ISBN - 0-7695-2367-6/05
10. Blair, P.: RFID Proving Ground Is All the World's Stage, METRO Group's (2007) (accessed January 24, 2008),
<http://www.rfidproductnews.com/issues/2007.09/metro.php>
11. Floerkemeier, C., Lampe, M.: Issues with RFID Usage in Ubiquitous Computing Applications. In: Ferscha, A., Mattern, F. (eds.) Pervasive 2004. LNCS, vol. 3001, pp. 188–193. Springer, Heidelberg (2004)
12. Banks, J., Hanny, D., Pachano, M.A., Thompson, L.G.: RFID Applied, pp. 311–318. John Wiley & Sons, Inc., Hoboken (2007)
13. Bacheldor, B.: Strong sales growth expected for RFID tags, Manufacturers' Monthly, December 10 (2007) (accessed February 11, 2008),
http://www.manmonthly.com.au/articles/Strong-sales-growth-expected-for-RFID-tags_z138655.htm
14. Lee, S.: Mutual Authentication of RFID System using Synchronized Information, M.Sc.Thesis, School of Engineering, Information and Communications University, South Korea (2005)
15. Sybase, Inc., SYBASE RADIO FREQUENCY IDENTIFICATION (RFID) TECHNOLOGY ARCHITECTURE, One Sybase Drive Dublin CA, 94568 USA (2005) (accessed February 11, 2007),
http://www.sybase.com/sb_content/1031464/16056_RFID_Arch_L02607_FNL3.pdf
16. GAO RFID, RFID Solutions for Healthcare Industry (accessed February 10, 2007),
<http://healthcare.gaorfid.com/>
17. U.S. Government Accountability Office.: Radio Frequency Identification Technology in the Federal Government, 441 G Street NW, Room LM Washington, D.C. 20548 (2005)
18. Chao, C., Jen, W., Chi, Y., Lin, B.: Improving patient safety with RFID and mobile technology. International Journal of Electronic Healthcare 3(2), 175–192 (2007)
19. van der Togt, R., van Lieshout, E.J., Hensbroek, R., Beinat, E., Binnekade, J.M., Bakker, P.J.M.: Electromagnetic Interference From Radio Frequency Identification Inducing Potentially Hazardous Incidents in Critical Care Medical Equipment. JAMA 299(24), 2884–2890 (2008)

Communities of Practice and Semantic Web Stimulating Collaboration by Document Markup

Christine Müller

Department of Computer Science, University of Auckland, New Zealand
Computer Science Department, Jacobs University Bremen, Germany
c.mueller@jacobs-university.de
<http://kwarc.info/cmuller>

Abstract. We believe that mathematics is the language of science and has paved the way of many innovations. However, mathematical research is often said to be “non-practical” and “hard to digest”. Furthermore, experts have access to highly specialized results, but are often less aware of applications outside their own community.

Our work promotes the exchange of *highly-specialized knowledge* between individuals with different mathematical background. We draw on modern *representation formats* to mark up structure and meaning of mathematical texts to reduce *barriers of communication*. Moreover, we integrate modern web technologies to build an adaptive, active, and collaborative environment, in which users engage in a *community of practice*.

Keywords: content markup, communities of practice.

1 Introduction

Mathematics is one of the oldest disciplines and the basis for most scientific and industrial innovations; mathematicians have paved the way to new scientific inventions allowing others to develop mathematical methods further and to provide a practical use. However, mathematical research is often said to be “non-practical” and “hard to digest” as many struggle to interpret mathematical writings. Furthermore, mathematical experts have access to highly specialized results, but are oftentimes less interested and aware of applications outside their own community. Although interdisciplinary collaborations increased over the years, most communication of mathematical knowledge is based on documents that solely present the solutions of problems; mathematical publications do not provide detailed insight in methods, deadlocks, and practices involved in the problem solving process or further examples and illustrations. Oftentimes, mathematical results are only partial articulation; many details are left out and only reside in the expert’s mind. Professional mathematicians can rediscover the missing steps and resolve ambiguity by drawing on their mathematical background and experiences, while less experienced readers and novices to the respective mathematical field may struggle.

Our work promotes the exchange of *highly-specialized knowledge* between individuals with different mathematical background by improving the access to

technical documents as well as stimulating online collaborations. We draw on *mathematical content markup formats* to mark up structure and meaning of technical texts to reduce *barriers of communication*. Moreover, we integrate modern web technologies to build an adaptive, active, and collaborative environment, in which users engage in a *community of practice*.

2 Methods

Communities of Practice (CoPs) [24] are groups of people who share an interest in a particular domain. By interacting and collaborating around problems, solutions, and insights they develop a shared practice, i.e. a common repertoire of resources consisting of experiences, stories, tools, and ways of addressing recurring problems. The theory assumes that learning is a *collaborative activity* rather than the reception of factual knowledge or information. By participating in a community of practice novices gradually increase their engagement with the community and expertise, while moving from the outside towards the inside.

Content-Oriented Representation bridge (fully formal) mathematical input languages and presentation markup languages. Formal languages, such as CASL [1], enforce full formalization of knowledge but provide sophisticated services, such as automatic program verifications or proof checking. In contrast, presentation-oriented languages, such as T_EX or L^AT_EX, provide informal and flexible ways to express knowledge concepts but only facilitate limited machine support such as type-setting or keyword search. Content-oriented representation formats for mathematics, such as MATHML [23], OPENMATH [17], sT_EX [11], or OMDOC [10], do not enforce full formalization but are more tedious to write than presentation-oriented languages. Content markup *explicitates the structure and meaning of documents*, which does not facilitate automatic proving or verification but much stronger services than informal formats (see Sect. 5). This work builds on the XML-based, web-scalable *Open Mathematical Document Format* (OMDOC [10]), which serves as *content markup format* and *ontology language* for mathematical documents on the World Wide Web.

3 A Case Study on Community of Practice

The theory of *community of practice* allows us to focus on the *social* and *practice-oriented* nature of mathematics. Despite a common belief that mathematical practitioners prefer isolation and self-study, we observed that they are *highly collaborative* and *active in their community*. Mathematical collaborations are essential for any stage of mathematical practice: from identifying a challenging problem, becoming acquainted to new problem-solving methods, up to the verification of results and peer-review of publications.

We also observed that mathematics is divided into several *heterogeneous* sub-communities. Although outsiders may get the impression that mathematical practitioners form a homogeneous, unified community and share the same practices all over the world, they actually form various sub-communities that differ

in their *preferred notations*, *basic assumptions*, and *motivating examples*. We believe that different mathematical practices hamper understanding and constitute *barriers of communication*.

Our case studies are a mixture of literature analysis, student group discussions, and interviews with professional mathematicians of various disciplines. For a general understanding of mathematical practice we drew on literature such as [8,4,3,19,21]. Further research focused on specific practices such as the choice of mathematical notations [22], basic assumptions and logical foundations [20], and the choice of typical examples [7]. We have gained intuitions on notations based on a web survey and an analysis of our Computer Science course materials with student volunteers (see [14]). We are currently conducting interviews with experts from various mathematical fields (including randomness, complexity theory, computability, and group theory) to gain insights in potential add-on services throughout the document life cycle, i.e. the writing, review, publication, search, and finally study and reuse of mathematical publications.

4 Novel Techniques for Technical Communications

Based on our observations in Sect. 3, we distinguish two types of services: Services for *reducing barriers of communications* (Sect. 4.1) and services for *stimulating technical collaborations* (Sect. 4.2).

4.1 Services for Reducing Barriers

We believe that different mathematical practices, such as mathematical notations and background assumptions, hamper communication, even among professional mathematicians. For example, C_k^n (Russia), C_n^k (France), and $\binom{n}{k}$ (Germany) denote the same concept, while $\mathbb{N}$ can be defined as *the set* $\{1, 2, 3, \dots\}$ (positive integers in number theory) or *the set* $\{0, 1, 2, 3, \dots\}$ (non-negative integers in set theory and computer science). The first example can cause unnecessary misunderstandings as users are actually referring to the same concept (the binomial coefficient) but know different presentations. The second example can cause pitfalls as users may believe to talk about the same concept but actually define it differently. Not knowing the proper constraints and side effects of a concept can cause inconsistent reuse as well as error-prone applications.

Markup of the *implicitly inscribed* structure and meaning allows us to *explicate practices* involved in the production of mathematical texts, potentially, reducing *ambiguities* and *barriers*. In the following, we illustrate how the markup of mathematical notations can facilitate the adaptation of alternative presentations and increase the *accessibility of documents* for the (novice) reader.

The structural markup of mathematical notation is specified in the two web-standards, OPENMATH and MATHML, based on which we specified a framework for the context-aware configuration of mathematical notations [9]. In [16] we have extended our approach with user (and community) modeling techniques to facilitate system-driven adaptation based on the readers' profiles and situations. The notation framework has been implemented in the Java library JOMDOC [6] and

will be integrated in our web-based reader *panta rhei* (see Sect. 5). During the adaptation of notation JOMDOC can enrich technical documents with interactive triggers, which can be interpreted by the Javascript Framework JOBAD [5] to provide an (inter)active reading environment.

The study of mathematical notations has revealed insights into other scientific and industrial scenarios, which we can now better understand and support. For example, consider a *needs requirements life cycle*: Specifying what a project should accomplish can be challenging. It's easy to misunderstand requirements if they are not clearly articulated. We do not claim that we can manage all factors of the requirements gathering process, but we can reduce some of the pitfalls. For example, based on the markup of technical concepts, we can explicate whether a client, salesman, or programmer refer to the same requirements or whether they denote different concepts similarly. Vice versa, we can identify whether varying labels actually refer to the same knowledge concept.

4.2 Stimulating Technical Cooperation

Adaptation of documents and explication of meaning can reduce misunderstandings, but do not necessarily stimulate cooperation and learning, which is defined as a *collaborative activity* that requires participation in a *community of practice* (see [24]). Traditional mathematicians prefer face-to-face collaborations with personal brainstorming and discussion using blackboard and chalk. Nevertheless, with the increase of global collaborations, more and more mathematicians draw on modern information technology. However, online communication is still tedious due to the nature of the mathematical language, a mix of formal notations and natural language text (see [16]). Mathematicians thus call for more efficient discussion and collaboration spaces as well as review and publication facilities.

5 Implementation

In [15], we introduced the interactive and collaborative reader *panta rhei* [18], which facilitates readers to challenge, discuss, and rate online documents. Based on the sTeX-2-OMDOC-2-XHTML workflow, authors can write their documents in their preferred L^AT_EX editor (using the semantic sTeX package [11]) and publish their results in the web reader: L^AT_EX_{ML} [13] provides the conversion from sTeX to OMDOC, while XSLT stylesheets [10] facilitate the transformation from OMDOC to XHTML. During the conversion semantic identifiers and meta-data (the content markup) are preserved, which improves the web-accessibility of the imported documents. Markup of narrative structure allows us to adapt the size and navigation of documents (see [12]), markup of concepts allows semantic search and easy cross-linking, enhancement with action triggers facilitate interactivity (see [5]), while distinction of content and form supports different visualizations, such as the adaptation of notations (see [9]).

The *panta rhei* system integrates the imported contents with easy-to-use web facilities. Displayed documents are “read only”, but all users can input their

opinions, questions, and answers via a forum and annotation facility. Moreover, implicit and explicit user modeling techniques are applied to personalize the adaptation of content (see [16] for details). Fig. 1 displays the two constituents of the system: *panta* (the user interface) and *janta* (the backend service). *panta* implements the discussion, annotation, and tagging facilities and gathers information on the user's notation preferences, while *janta* takes over all the content and user data handling as well as adaptation (please see the *panta rhei trac* [18] for details).

The usability of the *panta rhei* system will be evaluated based on the *Cognitive Dimensions approach* [2].

6 Conclusion and Outlook

Our work promotes the exchange of technical knowledge by reducing barriers of communication and by providing a web-based collaboration space. We build on the semantic representation format OMDoc, which marks up the structure and meaning of mathematical knowledge and improves its searchability, display, and personalization on the Web. We integrate the rich representation layer (of OMDoc material) with discussion, interactivity, and animation facilities to provide an adaptive, active, and collaborative environment, in which users engage in communities of practices to exchange technical knowledge more efficiently. We are developing a prototypical implementation to illustrate and evaluate our approach. Both, conceptual work and implementation, are still on-going. We will soon release a proof-of-concept prototype of the *panta rhei* system, based on which, we can start our evaluation.

Acknowledgments. We would like to thank Andrei Aiordachioaie, Stefania Dumbrava, Josip Dzolonga, Jana Giceva, Michael Kohlhase, Christoph Lange, Darko Makreshanski, Dimitar Misev, Normen Müller, Alen Stojanov, and Jakob Ücker for their contribution to the *panta rhei*, JOMDoc, and JOBAD project. This work was supported by JEM-Thematic-Network ECP-038208.

References

1. CoFI (The Common Framework Initiative). In: Mosses, P.D. (ed.): CASL Reference Manual. LNCS. IFIP Series, vol. 2960. Springer, Heidelberg (2004)
2. Green, T., Peter, M.: Usability analysis of visual programming environments: A cognitive dimension framework. *Journal of Visual Languages and Computing* 7, 131–174 (1996)

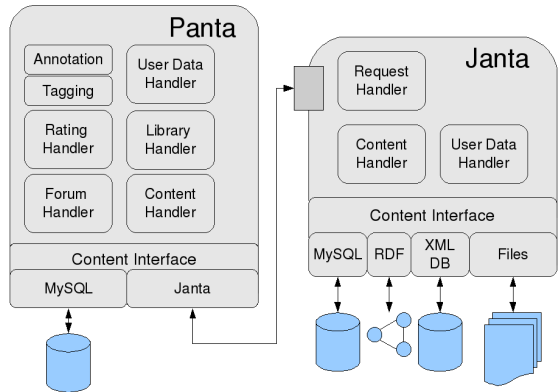


Fig. 1. System Architecture

3. Hardy, G.H.: *A Mathematician's Apology*. Cambridge University Press, Cambridge (1992)
4. Heintz, B.: *Die Innenwelt der Mathematik. Zur Kultur und Praxis einer beweisenden Disziplin*. Springer, Wien (2000)
5. JOBAD framework – JavaScript API for OMDoc-based active documents (2008), <https://jomdoc.omdoc.org/wiki/JOBAD>
6. JOMDoc Project (2008), <http://omdoc.org/jomdoc>
7. Kerber, M., Melis, E., Siekmann, J.: *Analogical reasoning with typical examples*. SEKI-Report SR-92-13, Universität des Saarlandes (1992)
8. Kitcher, P.: *Mathematical naturalism*. In: Kitcher, P., Aspray, W. (eds.) *History and Philosophy of Modern Mathematics*, pp. 293–325. University of Minnesota Press (1988)
9. Kohlhase, M., Müller, C., Rabe, F.: *Notations for Living Mathematical Documents*. In: Autexier, S., Campbell, J., Rubio, J., Sorge, V., Suzuki, M., Wiedijk, F. (eds.) *AISC 2008, Calculemus 2008, and MKM 2008*. LNCS (LNAI), vol. 5144, pp. 504–519. Springer, Heidelberg (2008)
10. Kohlhase, M. (ed.): *OMDoc – An Open Markup Format for Mathematical Documents [version 1.2]*. LNCS (LNAI), vol. 4180, pp. 25–32. Springer, Heidelberg (2006)
11. Kohlhase, M.: *sTeX: Using TeX/LaTeX as a semantic markup format*. *Mathematics in Computer Science. Special Issue on Management of Mathematical Knowledge* (in press, 2008)
12. Kohlhase, M., Müller, C., Müller, N.: *Documents with flexible notation contexts as interfaces to mathematical knowledge*. In: Libbrecht, P. (ed.) *Mathematical User Interfaces Workshop* (2007)
13. Miller, B.: *LaTeXML: A LaTeX to xml converter*. Web Manual seen (September 2007), <http://dlmf.nist.gov/LaTeXML/>
14. Müller, C.: *A Survey on Mathematical Notations*. KWARC report, Jacobs University Bremen (2008)
15. Müller, C., Kohlhase, M.: *Panta rhei*. In: Hinneburg, A. (ed.) *LWA Conference Proceedings*, pp. 318–323 (2007)
16. Müller, C., Kohlhase, M.: *Context Aware Adaptation: A Case Study on Mathematical Notations*. Research reports, Centre for Discrete Mathematics and Theoretical Computer Science, University of Auckland (November 2008)
17. OPENMATH Home. seen (March 2007), <http://www.openmath.org/>
18. The panta rhei Project (2008), <http://trac.kwarc.info/panta-rhei>
19. Polya, G.: *How to Solve it*. Princeton University Press, Princeton (1973)
20. Rabe, F.: *Representing Logics and Logic Translations*. Ph.D thesis, Jacobs University Bremen (2008)
21. Ruelle, D.: *The Mathematician's Brain*. Princeton University Press (2007)
22. Smirnova, E., Watt, S.M.: *Notation Selection in Mathematical Computing Environments*. In: *Proceedings Transgressive Computing 2006: A conference in honor of Jean Della Dora (TC 2006)*, Granada, Spain, pp. 339–355 (2006)
23. W3C. *Mathematical Markup Language (MathML) Version 2.0 (Second Edition)* (2003) Seen (July 2007), <http://www.w3.org/TR/MathML2/>
24. Wenger, E.: *Communities of Practice: Learning, Meaning, and Identity*. Cambridge University Press, Cambridge (2005)

Conceptualizing Software Life Cycle

Sabah S. Al-Fedaghi

Computer Engineering Department
Kuwait University
P. O. Box 5969 Safat 13050
Kuwait
sabah@alfedaghi.com

Abstract. A model of the software life cycle shows a software development process that includes all the activities and products required to develop a software system. This paper introduces a new approach to the specifications of the software life cycle, based on artifacts flow. The waterfall model—enhanced with feedback—is used as a sample of this flow-based methodology. Each phase of the development cycle is represented by five stages of the stream of things that flow. The resultant schema is a high-level abstraction of the software life cycle that enhances the specifications for development phases.

Keywords: Software development, life cycle, waterfall model, flow model.

1 Introduction

Modeling the software life cycle typically divides software development activities into distinct and ordered phases, starting from conception and ending with delivering the software product. The description includes many features, such as what is input and what is delivered by each phase, the transition criteria, feedback mechanisms, reviews, participants, and artifacts. There are several advantages to a life cycle model of software development. It provides a means for ordering and structuring the multi-faceted software construction process. Because of the nature of such a process, the model outlines concepts and relationships involved in converting real-life requirements into fixed instructions suitable for computer operations. Such a model also organizes the questioning and interpretation of actual manual procedures that are diverse in nature and purpose into methodologies that lead to systemic progress toward a finished software project.

This paper focuses on our conceptualization of software developed on the flow of artifacts we call *flowThings*. “FlowThings” refer to “software development things,” including temporary usable(s) and deliverables. They are tangible products and by-products that are produced and exchanged during each phase of software development. FlowThings are received, processed, created, released, and communicated. They are what flow between phases.

These deliverable flowThings appear in every project life cycle model. According to Patzak, “The starting point for the analysis of the phenomenon PROJECT is to look at a process ... In state O all more or less intended outcomes of the process ‘project

execution' are available, having been produced during the whole process. These *outputs* are *concrete* (products, organizations, etc.) or *abstract* (plans, knowledge, experiences, emotional states, etc.) or both" [15]. Outputs of these flowThings typically do not play explicit roles in specifications of life cycle models. For example, in the software waterfall model, the elements constituting the *fall* are rarely mentioned. In our search into this aspect, we found that one early model of the project management process that explicitly represents *deliverables* is Kapur's information system project life span, partially illustrated in figure 1. The triangles between phases represent the specific deliverables shown in the boxes immediately below [20].

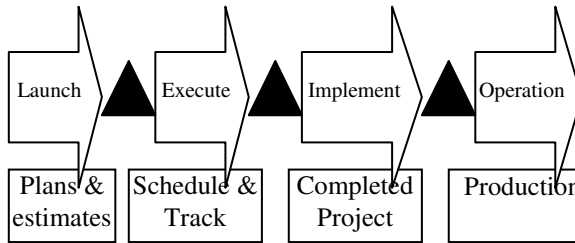


Fig. 1. Kapur's information system project life span. The triangles denote deliverables (from [20]).

A phase in the life cycle of software development cultivates deliverables or artifacts or, as we call them, *flowThings* that characterize the phase. For example, the requirement phase deals with "requirement things," such as specifications that include user stories, requirement specification reports and documents, user cases, different types of diagrams and cards, etc. In the implementation phase the "implementation things" are artifacts that include source and object programs, system use cases, software interfaces, databases, etc.

We view the software life cycle (SLC) as streams of flow of artifacts or "SLC flowThings." Some of these SLC flowThings may be limited within a phase while others cross to other phases, resulting in "requirement flowThings," "design flowThings," "implementation flowThings," and so forth. A certain type of flowThings may trigger the flow of another type of flowThings.

This model represents a grand picture of flows of tangible artifacts within the SLC. The flow is the main character in this SLC model where the focus is on tangibility and flow-ness. We concentrate on the waterfall model in order to focus our discussion, though a great deal of the analysis is applied to other SLC models.

2 The Flow Model

The flow model (FM) has been used in several applications, including communications, human/machine interactions, and engineering requirement analyses [1, 2, 3]. In FM, the flow of "things" indicates movement inside and between spheres. The *sphere* is the environment of the flow, such as SLC phases and sub-phases. Each sphere

includes five stages, and each sub-sphere has its own five-stage schema. A sphere with its dynamic stream of flowThings forms a *system*.

FM is a model of *things that flow*. To use a neutral term, we will use the term *flowThing* to denote these things that flow. FlowThings has five states, as shown in figure 2.

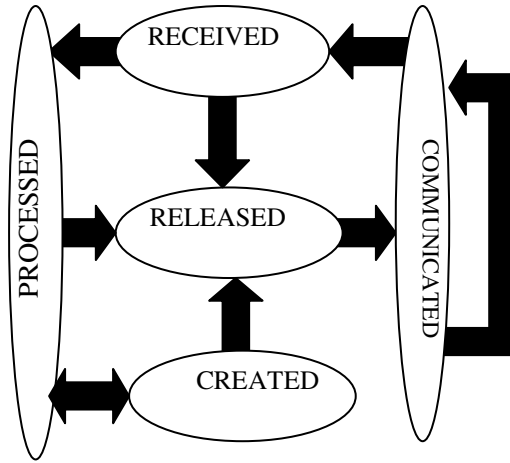


Fig. 2. Transition states of things that flow

The figure shows a dynamic movement (flow) of flowThings inside and outside the system. For example, a piece of information (flowThing) is created by company X, released, and communicated to another company, Y. In the system of company Y, the piece of information is communicated, received, processed, released, and then communicated to another company, say, Z. Thus, the information flows in this sequence of states:

In X: creation, release, communication (crossing to Y) .

In Y: communication, receiving, processing, releasing, and communication (to Z).

The flow structure is defined in terms of these five stages (referred to as a *schema*). Let us designate the stages of a flow schema as follows: REC is the receiving stage, PRO is the processing stage, CRE is the creation stage, REL is the releasing stage, and COM is the communication stage. An FM system, S, is defined as follows:

(1) S is a flowThing system such that $S \vdash \{REC, PRO, CRE, REL, COM, [S]\} \mid S$ [S] denotes a series of sub-systems (e.g., a company has several departments), and flows among stages defined as:

$REC \rightarrow PRO, REC \rightarrow REL, PRO \rightarrow CRE, PRO \rightarrow REL, CRE \rightarrow PRO, CRE \rightarrow REL, REL \rightarrow COM.$

where arrow $\rightarrow$ denotes flow of flowThings. $\vdash$ and $\mid$ denote a production process where several FM systems (with sub-systems produced in [S]) form one global system.

(2) Let f be the flowThing in system S1 and g be the flowThing of system S2, then:

- If f is ontologically similar to g , then $\text{COM} \rightarrow \text{COM}$ in the schemata of $S1$ and $S2$ is permitted. That is, if the *flowThing* is of the same type (e.g., information), then the flow between two systems goes through the communication stages of these systems.
- If f is ontologically not similar to g , then only $\sigma \rightarrow \rho$ in the schemata of $S1$ and $S2$ is permitted, where $\rightarrow$ denotes the triggering of flow in another system, and σ and ρ are stages. That is, any stage in $S1$ can trigger any stage in $S2$.

The *flowThings* are characterized by being exchangeable (received and communicated), creatable, processable, and releasable. Exchangeability means that elements can be imported and exported from and to other systems. Creatability means that new elements can be generated by the system. Processability means that elements can be changed to different forms. Releasability means that elements can be designated to be exported outside the system.

In addition to the fundamental characteristics of flow in FM, the following types of possible operations exist in different stages:

1. Copying: Copy is an operation such that *flowThing* $f \Rightarrow f$. That is, it is possible to copy f to produce another *flowThing* f in a system S . In this case, S is said to be S with copied features, or, for short, *Copy S*. For example, any *informational* schema (*flowThings* are pieces of information) can be copy S , while physical schemata (*FlowThings* are materials) are non-copied S . Notice that in copy S , stored f may have its copy in a non-stored state. It is possible that copying is allowed in certain stages and not allowed in others. Creation is different than copying in terms of generating new *flowThings* of the same type. For example, information may be processed to generate new information (e.g., information about John might generate the new information that *John is guilty*).

2. Erasure: Erasure is an operation such that *flowThing* $f \Rightarrow \emptyset$, where $\emptyset$ denotes the empty *flowThing*. That is, it is possible to erase a *flowThing* in S . In this case, S is said to be S with erasure feature, or, for short, *erasure S*. Erasure can be used for a single instance, for all instances in a stage, or for all instances in S .

3. Canceling: Anti-*flowThing* f^- is a *flowThing* such that $(f^- \cup f) \Rightarrow \emptyset$, where $\emptyset$ denotes the empty *flowThing*, and $\cup$ denotes the presence of f^- and f simultaneously. It is possible for the anti-*flowThing* f^- to be declared in a stage, a schema, or a sphere. If *flowThing* f triggers the flow of *flowThing* g , then anti-*flowThing* f^- triggers anti-*flowThing* g^- .

An example of use of these FM features is that of erasing a *flowThing*, as in the case of a customer who orders a product and then cancels the order. This may require the cancellation of several flows in different schemata that were triggered by the original order. Copying is an important feature for some types of *flowThings*, such as information. For example, a received order may be stored in the receiving stage while its copy is passed on to the processing stage. Such a feature may be important in declaring constraints, as in the case of personal information privacy. Similarly, destroying (erasure of) information may be an operation that needs strict control.

In FM, a system denotes a dynamic movement (flow) of flowThings inside and outside different stages. The system structure is defined in terms of the five-stage schema described previously.

To complete our description of FM, we need to show how to represent such things as attributes and properties (e.g., the color red representing the concept of courage). In object-oriented modeling many of the items rejected as *objects* are properties of things. For example, consider a person having a language skill. Because a skill is not a tangible thing, it must be modeled as an attribute of the person. In UML, attributes of association class instances (“link objects”) are attributes that belong to a connection of two or more objects.

FM spheres are spheres of entities: shop, customer, vendor, and even of abstract entities, such as job. Entity type in FM depends on the stages that compose the entity. The following are examples of entities with partial five-stage/sub-stage schemas:

- Entities with no *processing* stage, such as dumb terminals
- Entities with no *creation* stage, such as inventories
- Entities with no *receiving* stage, such as clocks
- Entities without internal flow, such as information: numbers, literals; physical entities (we disregard the atomic level for the time being).
- Entities that have only a *communication* stage, such as electric wire.
- Entities that have only *creation*, *release*, and *communication* stages, such as a pure electric generator.
- Entities that have no *communication* stage, such as stand-alone devices.

Consider the notion of intrinsic and extrinsic property. In semantics, a *property* is a quality inherent in the meaning of a word; e.g., “young” is a semantic property of baby, kitten, colt, etc. The intrinsic property of a thing is a property that is essential to the thing’s identity. It is used to define a class. In modeling, it is often said that some properties, such as *color* and *length*, are common to all or some classes of objects. Some objects have properties that are specific to themselves. In UML, an object property may be modeled as a binary association between classes. In semantic Web languages such as RDF and OWL, a property is a *binary* relation; that is, it links two individuals or an individual and a value.

A property is defined in FM as *something that flows in entities*. Notice that each entity in FM has multiple spheres, just as a human being has several systems: digestive system, musculoskeletal system, nervous system, etc. Food does not flow in the nervous system, and obtaining oxygen is not the function of the digestive system. Similarly, a business information sphere is different from its physical sphere, which is different from its money sphere, etc.

Figure 3 shows a diagram of an isolated flow system in FM. The flowThing is assumed to be a *stage-less* system. Usually we do not show the flowThing node when we draw a system except when necessary. FlowThings are written inside the sphere as a shorthand to indicate that the flowThing is of that type (examples follow). Notice that, by definition, a system consists of things that flow; hence the flowThing is an integral part of the system.

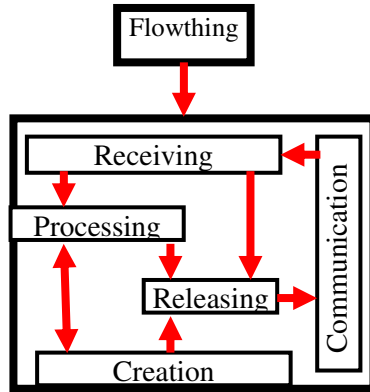


Fig. 3. An isolated system in FM

Thus, in figure 3, suppose that the system represents a human being. The flowThing *courage* denotes that this human being is courageous. Strictly, the model expressing that courage flows internally in the system.

3 Example of FM Modeling

Consider orders in an online ordering system as flowThings. Figure 4 shows a typical use case in the requirement phase of a transaction between Customer and Service provider. Using this example, we illustrate the notion of flowThings and show a sample of FM description.

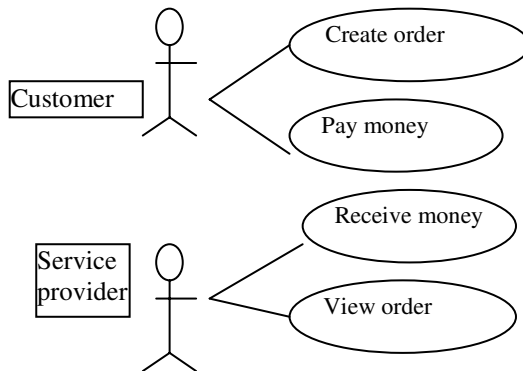


Fig. 4. Typical use case diagram

Figure 5 shows the FM description that corresponds to the *use case* in figure 4. The FM description includes two spheres: customer's sphere and provider's sphere. The customer's sphere includes three types of flowThings—order, invoice, and money.

The ordering transaction is represented as follows: The customer creates an order (circle 1) that flows to the provider's orders system (circle 2). The order is processed and triggers (circle 3 – dotted arrow) the creation of an invoice in the invoice system. The invoice flows (circle 4) to the customer's invoice system (circle 4), which triggers (circle 5) the creation of money (e.g., money order) that flows to the provider's money system. Comparing the FM description with the use case representation of figure 4, we can see that the FM modeling reflects a blueprint of the whole transaction using the five-stage schema repeatedly.

In this paper, we propose the use of such a modeling technique in all phases of software development. FM methodology serves as a description inside each module (e.g., use case), system (the five-stage schema), and sphere or collection of spheres, and across phases of the waterfall model.

To limit our demonstration of the FM's descriptive capabilities, we applied it to the waterfall model. According to Laplante and Neill [9], "In a survey of almost 200 practitioners, accounting for several thousands of projects over the past five years, the dominant process model reported was the [w]aterfall, with more than a third claiming its use."

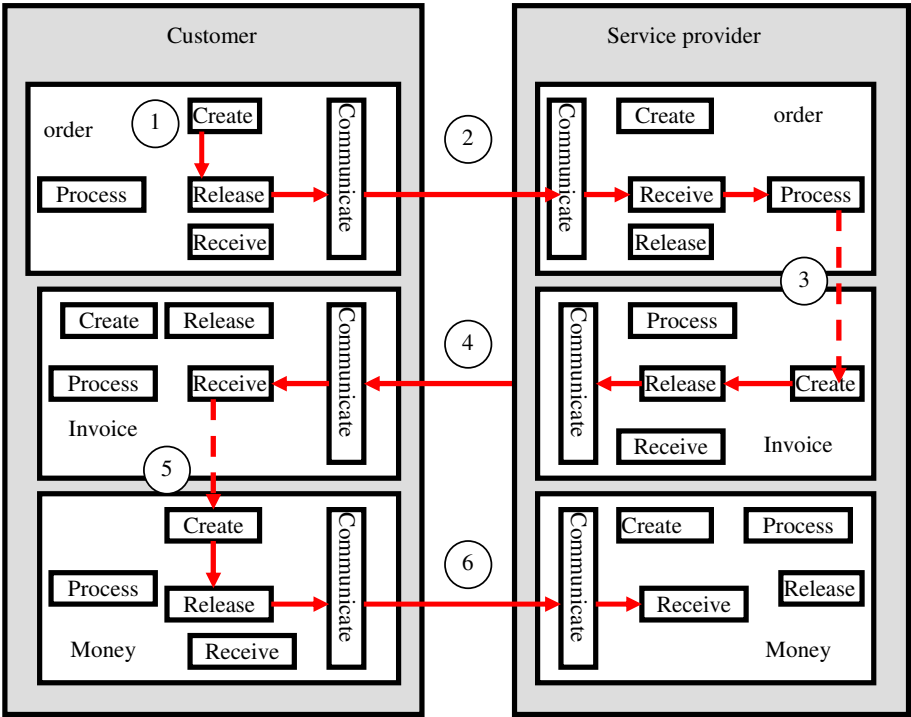


Fig. 5. FM description that corresponds to the use case in figure ?

4 Waterfall Model

In the waterfall model, software development is seen as flowing steadily downward through a sequence of phases. There are several variations of the model in terms of phases; however, the most common description divides the model into requirements analysis, design, implementation, testing, integration, and maintenance, as seen in figure 6. Development proceeds from the first phase of the requirements specification and then goes to design, where a blueprint is drawn for implementers. The implementation phase produces software components that are integrated into one system. Then the software product is tested, installed, and later maintained.

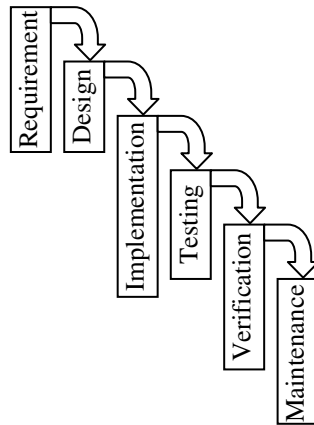


Fig. 6. The waterfall model

The model reflects a structured approach in which the development process progresses sequentially through well-understood phases with specified targets. Nevertheless, the original waterfall model has been criticized because it lacks iteration and feedback among phases, which are essential features in the software development process [11]. It is difficult to finish a phase completely before moving on to the following phases. “Many of the details only become known to us as we progress in the implementation. Some of the things that we learn invalidate our design and we must backtrack” [14]. According to Laplante and Neill [9].

The modern reality of software development is that change is unavoidable and must therefore be explicitly accommodated in the life cycle. It is not an error that must be fixed; it's a natural aspect of system construction. This change is not isolated to requirements, but the requirements example is the most immediate and most significant. The more we understand something, the more we realize the flaws in our initial assumptions and conceptions. If we cannot readily adapt our solutions to these changes, the costs of accommodating such requirements "errors" escalate exponentially.

Many models have been proposed to enhance the features of the waterfall model. Figure 7 shows a partial view of such a model used in developing safety-critical software [21].

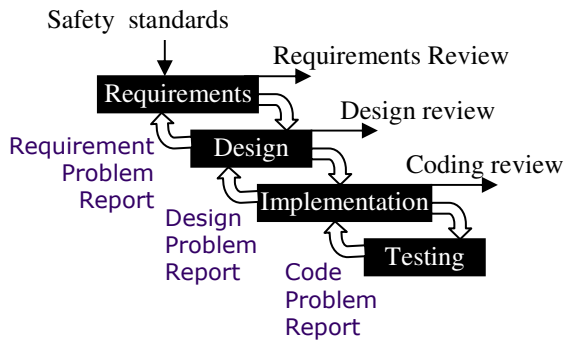


Fig. 7. Enhanced waterfall model

Alternative approaches exist for modeling the life cycle of software development. For example, the Spiral Design model breaks a software project into smaller projects with associated risks. After addressing major risks, the model terminates as a waterfall model with six steps:

- Determining objectives and constraints
- Identifying and resolving risks
- Evaluating alternatives
- Developing and verification
- Planning the next iteration
- Iterating

Extreme programming emphasizes certain methodologies, such as use developing, testing cases before coding, and putting most of the documentation into the code.

5 The Flow Model and the Waterfall Model

The basic architecture of the waterfall model represents a systematic order of phases in software development. The “fall” denotes a sequence of phase dependency. “The deliverables of one phase ‘fall’ into the next phase much like the waterfall metaphor used in the name of the methodology” [16]. So what is the nature of this “fall”? Apparently, it means finishing one phase and entering the next one. Rationality implies that identifying requirements precedes design, design precedes implementation, etc. This is reflected in the model by arrows that point out the sequence of phases. It is also rational to include iteration and feedback as basic ingredients in the enhanced waterfall model. So the “fall” (i.e., arrows) is nothing but the control flow seen in such things as flowcharts. When we finish the requirement phase, control moves to the design phase, etc. The “fall” may include returning control back to a previous phase.

If such an interpretation is acceptable, then a fundamental question is—what is the nature of this “control”? In computer science the flow of control refers to the order in which the instructions are executed. If this idea is applied to the waterfall model, then the phases are a type of control flow statement, or a type of function. Put in more accessible terms, phases are processes. “Processes” indicate activities that must be undertaken. Still, while the flow of control is well defined in programming languages through several mechanisms, such as formalized calls, parameters, local and global variables, etc., the interior and control flow between phases of the waterfall model are characterized by vagueness and generalities. Certainly, we cannot achieve the precision of programming languages in describing a complex process such as the SLC; nevertheless, it is possible to achieve a more precise description than the current specification of the waterfall model.

In this paper, we claim that we can achieve this precision through the following steps:

1. Applying the five-stage schema in every phase and sub-phase of the waterfall model.
2. Defining the “fall” in the waterfall model as the flow of flowThings.

6 Flow-Based Waterfall Model

In this section, we redesign the waterfall model on the basis of the notion of flow. The flow-based waterfall model divides software development into phases where each stage has its own “things that flow.” Figure 8 shows a general view of such a model. The arrows inside the phases are omitted to simplify the figure.

The figure includes the six phases shown in figure 6. Each phase now has a five-stage schema: receiving, processing, creation, release, and communication. The model phases may have sub-phases. If that is the case, then each sub-phase is also represented by its own five-stage schema. Immediately, we can see that we achieve additional internal control in the phase. The flowThings in each phase are classified into received, processed, created, released, and communicated. Communicated flowThings flow to other phases, indicated by dotted lines. Here, dotted lines mean either solid arrows (i.e., the passing flowThings are of the same type), or dashed arrows (triggering a flow in the target phase). For example, a program created in the implementation phase and passed to the testing phase is indicated by a solid arrow. This simply means that the flowThing program flows from requirement to testing. On the other hand, a flowchart created in the design phase might trigger the creation of a program in the implementation phase. In this case the arrow would be a dashed line. Of course, the implementer can look at the flowchart; however, from the managerial point of view, the flowchart belongs solely to the designer.

Our basic assumption is that all these “things” are things that flow.

- Requirements (called *reqthings*) are received, processed, created, released, and communicated.
- Design things, such as flowcharts (called *desthings*), are received, processed, created, released, and communicated.

- Implementation things, such as programs (called *impThings*), are received, processed, created, released, and communicated.
- Testing things, such as programs and data (called *tesThings*), are received, processed, created, released, and communicated.
- Verification things (called *verThings*) are received, processed, created, released, and communicated.
- Maintained things (called *maiThings*) are received, processed, created, released, and communicated.

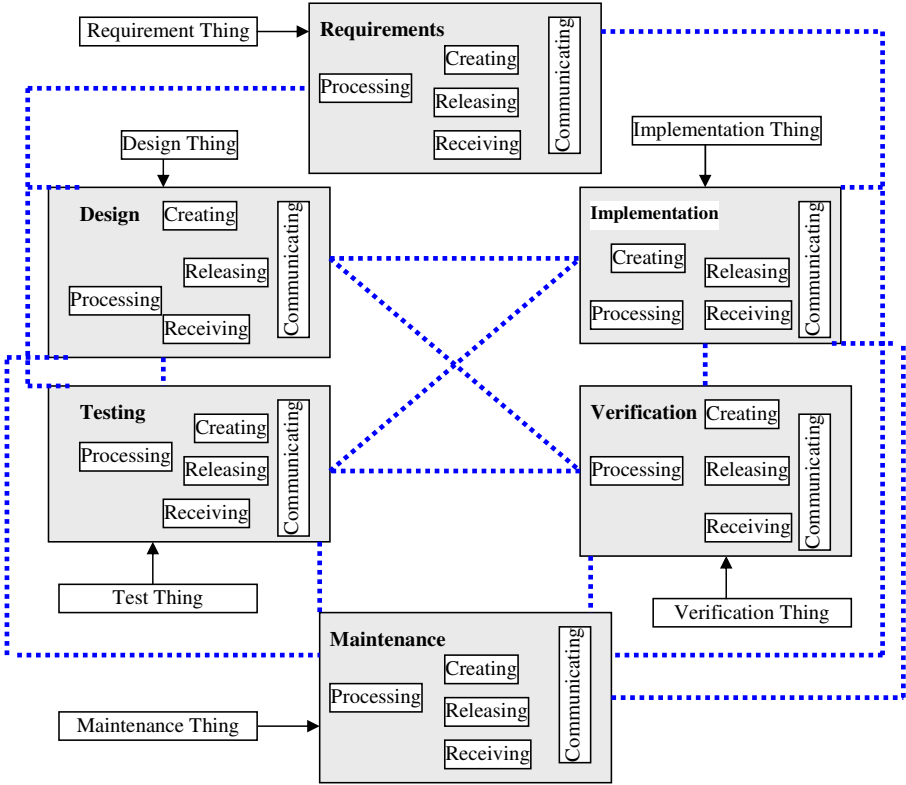


Fig. 8. A general view of a flow-based waterfall model

Because of space limitations, we apply the FM approach to the requirements and design phases. Other phases can be described in a similar way. The final objective of our research is to develop a software development system based on FM.

7 Requirements Phase

The things that specify requirements in terms of “requirement artifacts that flow” are denoted as *reqThings*. Consider a sample of *reqThings*: the user story used in Agile

Modeling [4]. “A user story is a very high-level definition of a requirement, containing just enough information so that the developers can produce a reasonable estimate of the effort to implement it” [4]. Ambler gives the following examples of user stories, denoted as set S:

- Students can purchase monthly parking passes online.
- Parking passes can be paid via credit cards.
- Parking passes can be paid via PayPal.
- Professors can input student marks.
- Students can obtain their current seminar schedule.
- Students can order official transcripts.
- Students can enroll only in seminars for which they have prerequisites.
- Transcripts will be available online via a standard browser.

The flow of these user stories is shown in figure 9.

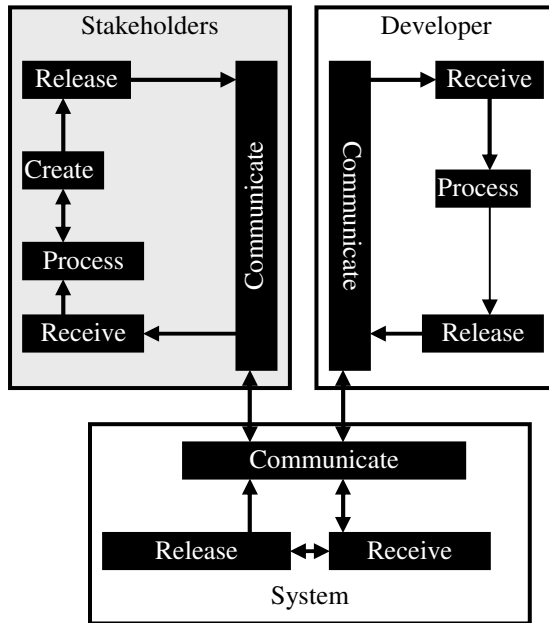


Fig. 9. ReqThings flow in the requirement phase

The system in the figure refers to the software development management system. Its role is assumed to be minimal and to include record keeping of the interaction between households and developers. Stakeholders create the reqThings and the developer processes them. The process is interactive until settling on the final set of user stories. Figure 9 presents a reqThing lifecycle model that tracks all activities related to creating, receiving, processing, and communicating user stories. A log of the reqThing model may be illustrated as follows:

1. User creates user story A1: Students can purchase monthly parking passes online via PayPal.
2. User story A1 released (stakeholder team finalizes it).
3. User story A1 communicated (information such as time, receiver id, etc., can be added).
4. User story A1 is communicated and received by developer.
5. User story A1 is processed by developer (proposal to split it is attached).
6. User story A1 is released to householder.
7. User story A1 is communicated to householder.
8. Householder receives and processes user story A1 including developer comments.
9. Two user stories are created, B1 and B2c, instead of the old user story A1:

B1: Students can purchase monthly parking passes online.

B2: Students can purchase monthly parking passes via PayPal.

10. B1 and B2 are released and communicated to developer, etc.

During this recorded interaction, other information can be added to user stories, such as priorities, estimated effort to implement, etc. At the end the user story is communicated to the design stage.

Notice that interaction is modeled in a systematic way according to the flow of the requirement artifact. The system can be automated to record every part of the flow at different stages. Structurally, all reqThings are created by householders because developers cannot create reqThings (creation stage is missing).

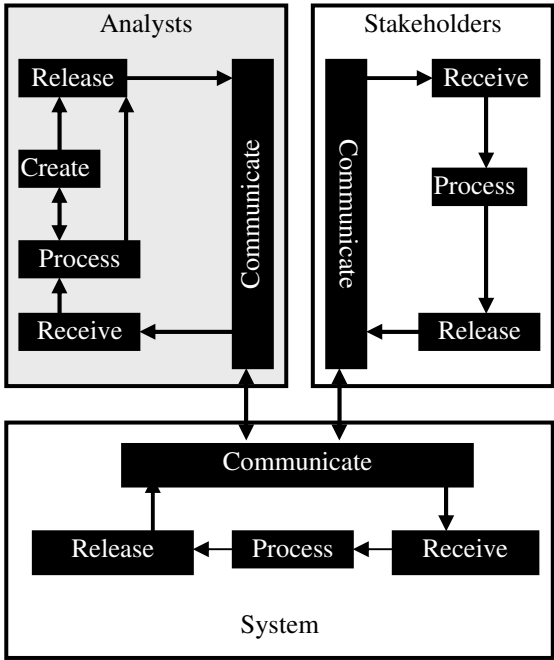


Fig. 10. AnaThings in the analysis phase

It is advantageous at this point to discuss similar low-level analysis activities: the concept of Agile Analysis complements the previously discussed interaction between householders and developers. It is distinguished from traditional requirements analysis because it is done in a highly collaborative manner. According to Ambler [5], Agile analysis is a process where developers and project stakeholders work together to understand the domain, to identify what needs to be built, to estimate functionality, to prioritize the functionality, “and in the process optionally produc[e] artifacts that are just barely good enough.” Artifacts in this phase can be called *anaThings*, and they include UML activity diagrams, class diagrams, constraint definitions, data-flow diagrams, E/R diagrams, flow charts, use cases, etc. Here the system is actively involved in the process.

Figure 10 shows process modeling of *anaThing* flow in this analysis. The creation of *anaThings* is the job of developers. In addition, the system participates in processing these *anaThings*.

However requirements and analysis are viewed, FM presents a suitable systematic apparatus for managing development of these requirements and analysis. Current methodologies describe these processes in general terms, while FM divides the phases into stages with logical connections between stages.

To show some of the advantages gained by this further categorization of the interiors of phases, we apply the method to the important field of traceability.

8 Requirements Traceability

Traceability is an important aspect of software engineering. A lot of research is done in the pre-traceability area. Requirements are basic factors in the architecture of the sub-systems of all phases, and the same is true for traceability. All specifications throughout the lifecycle of development have trails that lead back to the requirement phase. Usually, ontology is used to interrelate different artifacts for traceability purposes. Traceability in development phases is an essential systems development activity [13].

Traceability of artifacts in requirements and in later phases is known to be problematic because of the conceptual distance between requirements and technical specification [17]. Although “large systems should satisfy hundreds, even thousands of requirements, it is difficult to refine the architecturally relevant information contained in the requirements. [In addition], it is difficult to maintain the consistency and traceability between them since single requirements can map multiple architectural concerns and, contrarily, architectural components can have few relations to various requirements” [17].

Requirements traceability (RT) has been defined as “the ability to describe and follow the life of a requirement in both a forwards and backwards direction (i.e., from its origins, through its development and specification, to its subsequent deployment and use, and through all periods of on-going refinement and iteration in any of these phases)” [8].

Strašunskas presents a complete graph of traceability relationships among the first five phases of the waterfall model.

In such development process [waterfall method] the trace information should be captured, and management of those captured information is not really difficult. Only the final version of one stage will serve as input for next. So, the traceability should be established only between those fragments (e. g., requirements from the final requirement specification to concrete design) [17].

This approach to traceability in the software development process is illustrated in figure 11.

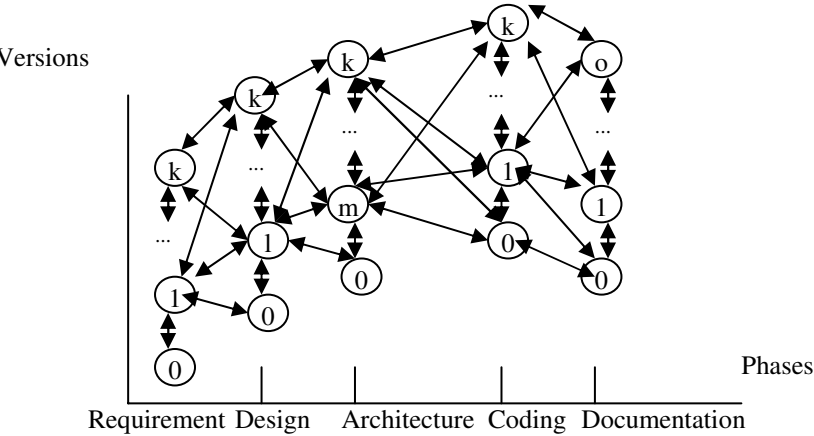


Fig. 11. View of traceability in a real interactive software development process (from [17] with modifications)

The traceability technique, integrated with version control and configuration management, can facilitate management of the composition of product fragments consisting of interrelated various model fragments, code fragments, and documents. The collaborative CASE tools should keep track of changes in different working modes [17].

FM is suitable for difference traceability techniques since phases of software development are divided into the five-stage schema with a clear type of flowThings. The flow of each flowThing can be traced by tracking the flow steps of its development. Suppose that the design phase triggers the implementation of program V. Program V is constructed from the locally coded program V' and the imported sub-program S. The high level stream of implementing V may be tracked from the point of its creation to its communication to the next phase, as shown in figure 12. First V is *created* through the *creation* of V' and *receiving* S. Both V' and S are *processed* and then merged to complete the *creation* of V. Different versions of V are processed with the possibility of retracting to previous steps. For example, when creating different versions of V, one might decide to change the imported module S, thus placing a new order for a different version of S. This is represented by the arrows labeled with circles 1 and 2 in the figure.

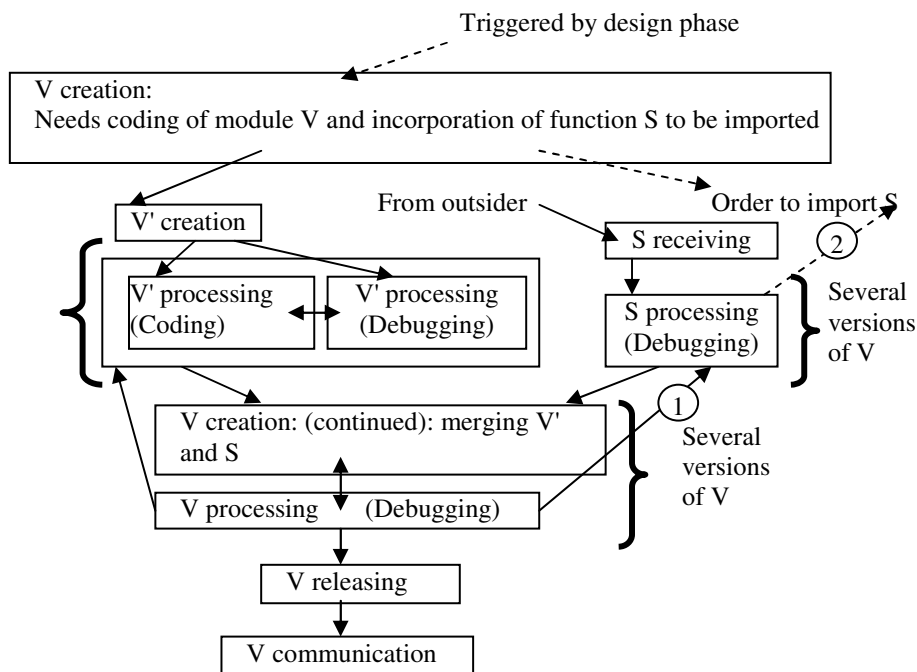


Fig. 12. Implementing V' from V and S . V' is created locally, while S is imported. Tracing in this scheme amounts to backtracking to previous versions of constituents.

We envision an automated system that traces the creation, processing, receiving, releasing, and communication of different flowThings and their constituents. In this case tracing the flow of flowThings is easier than lump-summing the flow stages in the phases of the waterfall model.

9 Design Phase

DesThings are flowing artifacts that specify design. For example, in business application development projects, desThings consist of computer-aided design (CAD) models, drawings, product design data, and other artifacts managed by systems such as PDM (product data management). UML diagrams may play the role of desThings in object-oriented technologies.

Software design is similar to the product design process. The product design process [7] is rich in specifications of such processes, with the aim of providing “more structure and a better way to manage the deliverables, resources and trade-offs” [10]. Hasenkamp gives the general scheme of a product design process, as shown partially in figure 13.

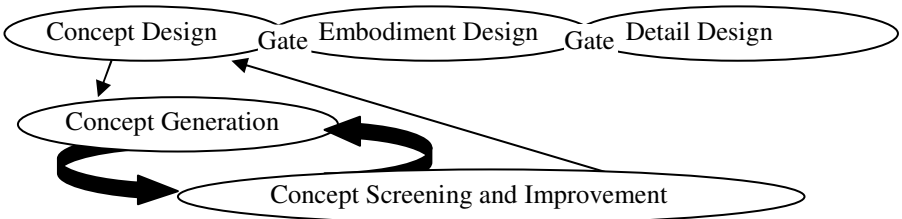


Fig. 13. Phases of a general product design process (from [7])

The Concept Design phase is divided into Concept Generation and Concept Screening & Improvement. According to Hasenkamp [7], gates allow for checking of the ongoing product design process. The concept generation phase results in rough design layouts—e.g., drawings and simple prototypes with key technical choices [Thornton [18] as referenced in [7]).

Notice that the two black arrows between Concept Generation and Concept Screening match the interaction between the creation and processing stages in FM. FM presents a more general model of the design process. Figure 14 shows the corresponding FM modeling of the design phase. Creation and processing are interrelated sub-stages in the five-stage schema. Receiving indicates that some desThings may be imported and incorporated into the design. Embodiment Design and Detail Design in Hasenkamp's figure (figure 13) are types of processing inside the processing stage. Thus, FM can provide a systematic description with uniform application of the five standard stages of flowThings.

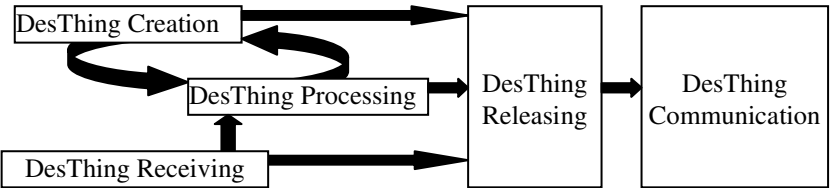


Fig. 14. FM modeling of the design phase

The product design process modeled in Hasenkamp's figure (figure 13) represents a community of researchers who work in this area (see references in [7]). Nevertheless, the whole approach lacks ontological clarity if applied to software products. The figure does not distinguish between desThings as tangible deliverables and abstract concepts. Hasenkamp [7] refers to several authors when he writes,

The Concept Design phase requires creativity, experience and skill of the designer ..., and merely applying appropriate tools and methods does not guarantee, e.g., more reliable products. Designers should be encouraged to take the customer's perspective and to think in terms of robustness... Supporting this, ... "concepts are often best generated by individuals", whereas the "concept

selection and enhancement is best performed in groups." However, to initiate and organize individual *thinking* processes, *brainstorming* sessions have proven to be a useful approach ...

Clearly, there are two ontological levels of things that flow: abstract concepts and their embodiment in some desThings, as shown in figure 15.

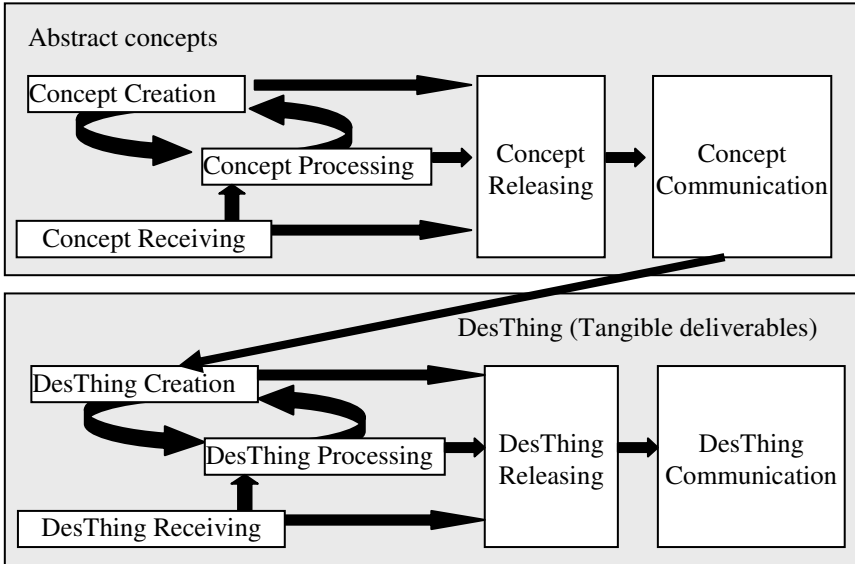


Fig. 15. Ontology of the design process

Individual *thinking* processes such as *brainstorming* belong to the upper sphere, while desThings can originate from such a sphere or be imported from the outside (receiving stage). Each level of ontology has its own management considerations. The (automated) desThings management system ought to register the sources involved in creating desThings (e.g., certain developers: consultants, stakeholders, management). FM, with its strict separation of streams of flowThings, presents a clear record of the product life cycle and all elements involved in the product's life.

10 Conclusion

This paper proposes incorporating a new model based on the notion of flow into modeling of the life cycle of software development. Each phase of the development cycle is represented by five stages in the stream of things that flow. The resulting description is a high-level abstraction of the SLC that enhances specification of development phases.

The new approach enhances the life cycle description in the following ways:

- Specification of generic stages of artifact flow inside each development phase of the life cycle provides more uniform details of the steps in software development.
- Application of the notion of flow across phases and sub-phases of the cycle gives a more uniform structure that may lead in the future to a formal description of the life cycle of software development.
- Our conceptualization of systems of things that flow (flowThings) seems to introduce a focus on features with wide application that can be used in other aspects of software development.

Though we have concentrated mainly on the waterfall model, enhanced with feedback, the flow-based methodology can be applied to other life cycle models. It has not been possible to produce a complete road map for all phases of the waterfall model. Nevertheless, concentrating on the requirements and design phases has provided an account of the FM methodology as applied in this area.

The main purpose of the paper is not to produce a complete description of a model of the life cycle of software development; rather, the objective is to raise interest in utilizing the flow-based approach. Several areas can be explored in this context, including further exploitation of FM itself, in addition to its application in life cycle modeling. We envision a software management system that will register all flow streams in and between the five stages of phases and sub-phases. "Book keepers" are stationed at every stage of the flow streams, recording information about flow activities. The requirements phase is a prime candidate for the first exploration of such a management system.

References

1. Al-Fedaghi, S.: Modeling Communication: One More Piece Falling Into Place. In: The 26th ACM International Conference on Design of Communication (SIGDOC 2008), Lisboa, Portugal, September 22-24 (2008)
2. Al-Fedaghi, S.: Informational Human-Machine Interaction. In: The 2008 IEEE International Conference on Systems, Man, and Cybernetics (SMC 2008), Singapore, October 12-15 (2008)
3. Al-Fedaghi, S.: Software Engineering Interpretation of Information Processing Regulations. In: IEEE 32nd Annual International Computer Software and Applications Conference (IEEE COMPSAC 2008), Turku, Finland, July 28 - August 1 (2008)
4. Ambler, S.W.: User Stories,
<http://www.agilemodeling.com/artifacts/userStory.htm>
5. Ambler, S.W.: Agile Analysis,
<http://www.agilemodeling.com/essays/agileAnalysis.htm>
6. Ambler, S.W.: Data Modeling,
<http://www.agiledata.org/essays/dataModeling101.html>
7. Hasenkamp, T.: Linking the Design Process with Design for Six Sigma, onesixsigma.com (May 10, 2007),
<http://www.onesixsigma.com/article/linking-the-design-process-with-dfss>

8. Gotel, O.C.Z., Finkelstein, A.C.W.: An Analysis of the Requirements Traceability Problem. In: Proceedings of the IEEE International Conference on Requirements Engineering (ICRE 1994), Colorado Springs, Colorado, April 18-22, pp. 94-101 (1994)
9. Laplante, P.A., Neill, C.J.: The Demise of the Waterfall Model is Imminent. and other Urban Myths. *Game Development Magazine* (February 2004),
<http://www.acmqueue.com/modules.php?name=Content&pa=showpage&pid=110>
10. Mader, D.P.: Design for Six Sigma. *Quality Progress*, 82-86 (July 2002)
11. McConnell, S.: *Code Complete*, 2nd edn. Microsoft Press (2004)
12. McConnell, S.: *Rapid Development: Taming Wild Software Schedules*, pp. 449-463. Microsoft Press, Redmond (1996)
13. Nuseibeh, B., Easterbrook, S.: Requirements Engineering: A Roadmap. In: Finkelstein, A. (ed.) *The Future of Software Engineering. Special Issue 22nd International Conference on Software Engineering*. ACM Press, New York (2000)
14. Parnas, D.L., Clements, P.C.: A Rational Design Process: How and Why to Fake it. *IEEE Trans. Software Eng.* 12(2), 251-257 (1986),
<http://users.ece.utexas.edu/~perry/education/SE-Intro/fakeit.pdf>
15. Patzak, G.: Model of Project Planning. In: Reschke, Y., Schelle, H. (eds.) *Dimensions of Project Management*, pp. 26-27. Springer, Berlin (1990)
16. Stamey, J.W.: TRIZ and Extreme Programming. *The TRIZ Journal* v1.0, 0.0 (2006),
<http://www.triz-journal.com/archives/2007/02/03/>
17. Strašunskas, D.: Traceability between Fragments throughout Lifecycle of Collaborative Systems Development. *Inf. Sci.* 24 (2003),
<http://www.leidykla.vu.lt/inetleid/inf-mok/24/etomas24.html>
18. Thornton, A.: *Variation Risk Management, Focusing Quality Improvements in Product Development and Production*. John Wiley & Sons, Chichester (2003)
19. Vashishtha, S.: Agility Meets the Waterfall, *JavaWorld.com* (March 25, 2008),
<http://www.javaworld.com/javaworld/jw-03-2008/jw-03-agile-practice.html>
20. Wideman, M.: The Role of Project Life Cycle (Life Span) in Project Management: Project Life Spans in the 1990s, February 1 (2004), (accessed, June 2008),
<http://www.maxwideman.com/papers/plc-models/1990s.htm>
21. Aonix, *Safety Critical Software Using Ada* (2008),
http://www.aonix.com/objectada_sc_handbook.html

Modeling Complex Adaptive Systems

I.T. Hawryszkiewicz

University of Technology, Sydney
igorh@it.uts.edu.au

Abstract. The paper describes ways to model systems that dynamically change as their environment changes. The generic term complex adaptive systems (Kovacs, 2005) is increasingly used to describe systems in such environments. Complex adaptive systems are generally defined (Holland, 1995) as made up of many agents (which may represent cells, individuals, firms, projects) acting in parallel, constantly acting and reacting to what the other agents are doing. The control of complex adaptive systems tends to be highly dispersed and decentralized. The overall behaviour of the system is the result of a huge number of decisions made every moment by many individual agents. Processes in such systems need to be equally adaptive and we refer to them as complex adaptive processes. Currently there are no widely accepted methodologies to model and design complex adaptive processes. Most methodologies for information systems design focus on prescribed processes. The paper describes ways to model such systems. The models will differ from existing modeling techniques as they combine business functions with social structures in ways that facilitate social connectivity and interactivity needed to adapt to changing situations within the business context. At the same time the social networks will be used to define the knowledge requirements that capture the outcome of work exchanges to support process continuity. It develops the idea of collaboration graphs to integrate social network into business models. It develops a blueprint based on three parts, business models, collaboration and knowledge and develops models based on integrating these three components. It then demonstrates the methods in outsourcing environments and principles of implementation of the models using contemporary technologies.

Keywords: Social Networks, Collaboration, Lightweight technologies.

1 Introduction

The growth of business value networks is creating demands for communication systems to support them. The growth of collaboration is calling for ways to use new communication technologies, especially those based on Web 2.0, to develop enterprise wide collaborative strategies to support enterprise wide collaboration. Such evolution is even more important with a trend to what are known as dynamic organizations. One definition of dynamic organizations is what is known as Enterprise 2.0. It is perhaps fair to say that Enterprise 2.0 sets a direction rather than a concrete structure. Enterprise 2.0 was introduced by McAfee (2006) in his article in the Sloan Management

Review as a natural trend towards obtaining additional competitive advantage by using the new technologies available through Web 2.0. It sees a business environment where collaboration extends from groups and individuals to organizational units and whole enterprises.

There is also a general view that technology can support and facilitate collaboration especially where knowledge sharing is critical. However, at the same time technology must be adapted to collaboration. This is still quite challenging. Most technical support systems are based on preprogrammed activities whereas collaboration is often emergent and requires support systems that can evolve as collaboration evolves. Furthermore, such changes must be carried out by knowledge workers themselves. The most common approach is to provide the communication tools and allow users to choose the most appropriate at any given situation.

There is however a growing trend to use a more systematic and strategic way to support enterprise wide collaboration including the formation of strategic communities (Kodama, 2005) and communities of practice. Enterprise wide collaboration is important where a number of activities have to be coordinated to achieve a particular goal or mission. This requires planning of work processes and maintaining context and awareness across many activities. It also requires the sharing on any created knowledge during the process among these activities.

This paper proposes that enterprises need to consider the development of what are called here collaborative infrastructures to get the benefits of collaboration. The most important aspect of such infrastructures is to support relationships between participants to share knowledge and to develop knowledge communities or hubs. This paper provides an approach for this purpose by defining collaboration graphs that are extensions of social network diagrams.

2 Collaborative Infrastructures

The ultimate goal is to develop enterprise architectures like that in Figure 1 to support knowledge workers (Davenport, 2005). The major components in Figure 1 are:

- The participants, who in our case are mainly knowledge workers, and the human resource policies used to define their responsibilities,
- The ERP systems and activities that are repositories of everyday transactions. These are usually structures and follow well-defined processes and the software is provided by vendors.
- The analytical activities that support innovation and are flexible in the sense that they develop new ideas or solve existing problems,
- The collaborative infrastructure that allows knowledge workers to collaborate and create the new knowledge. This is implemented using lightweight workspaces,
- Any new knowledge created by the work of knowledge workers. This can be captured using Web 2.0 technologies.

Processes in such environments usually emerge rather than being preplanned.

Complex adaptive processes are currently not well-defined in any formal manner. Our challenge is to define the special characteristics of adaptive processes and provide

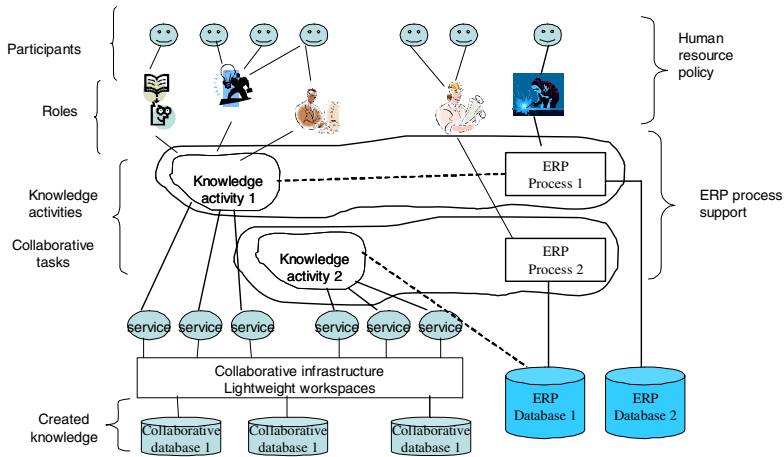


Fig. 1. An Integrated Enterprise

ways to design them. Complexity theory provides guidelines for defining complex adaptive processes. This paper draws on aspects of complexity theory as defined (Merali, McKelvey, 2006) and that of complex adaptive systems (Holland, 1995). The criteria here include:

- The ability to self organize at local levels in response to a wide variety of external changes,
- The creation and quick establishment of self contained units that address well defined parts of the environment,
- Loose coupling between system elements and a control system to reorganize the structure to respond to external change,
- Ability to organize connections between units and support the changed connections and interactivity.
- Aggregate smaller units into larger components with consequent changes to the connectivity and interactivity,
- Realization of simple interfaces between model components.

The impact of this trend is two fold, namely:

- Design methodologies must be able to cater for the dynamic nature of processes and include specific criteria in modeling that emphasize such dynamic nature. Hence activities must be modeled as independent and loosely connected entities, and
- Technical solutions must support user driven change, which is referred to as lightweight technologies in this paper.
- Social structures and relationships play a key role and must be considered on a par with business activities.

This paper describes modeling methods that for complex adaptive systems and convert the models to support systems, which facilitate the work of knowledge workers in complex environments.

3 A Blueprint for Modeling

From the perspective of design theory (Gregor, Jones, 2007) the method described in this paper proposes a central blueprint for modeling complex adaptive systems. The blueprint is shown in Figure 2, and combines business activities, collaboration graphs, and knowledge requirements as the three basic constructs for any model. It is called a blueprint as it sets the guideline for all modelling levels. It requires all modeling to combine the business activities with collaborative networking and sees networking as a link between the different activities. Similarly knowledge requirements include the explicit databases found in most business systems. They also include records of collaborative interactions integrated into the activities. Collaboration networks are also important as they indicate the collaborative knowledge requirements.

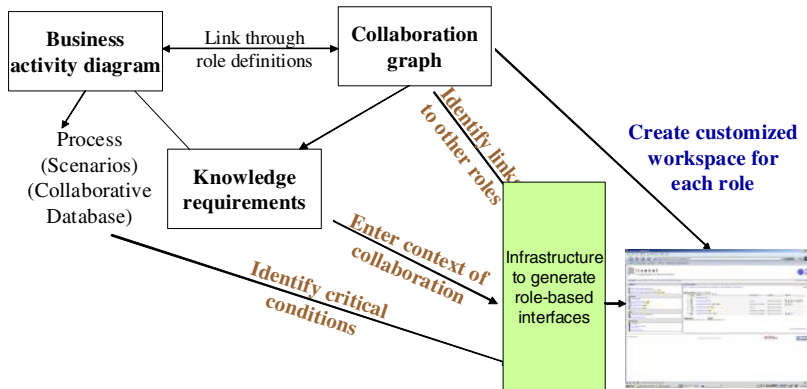


Fig. 2. The blueprint for modelling adaptive information systems

The model components will then be mapped to technologies. The mapping will first define an infrastructure that can be used to generate workspaces derived from the collaboration graph and the business activities. The collaboration is thus placed in the business context. It will then generate role based workspaces that can be dynamically changed as a situation evolves. All process components and relationships relevant to a role are placed into the one workspace space thus placing collaboration in the business context. Our research provides systematic ways to convert models to an implementation as the form and function in Gregor and Jones (2007).

4 Modeling Activities

We describe the modelling method using an outsourcing example in Figure 3. At the top-level it shows four organizations. The process service provider maintains a service

(which may include a number of applications) to a client and subcontracts the provision of application programs for a third party, the application vendor. At the same time there is a network provider, who supplies the network and any operating systems to support the outsourcing arrangement. Different roles are associated with each of these organizations. People assigned to these roles must collaborate to resolve any issues. In this case the initial analysis indicates a business requirement to maintain a quality of service to the client through response to queries and general maintenance of a level of client satisfaction.

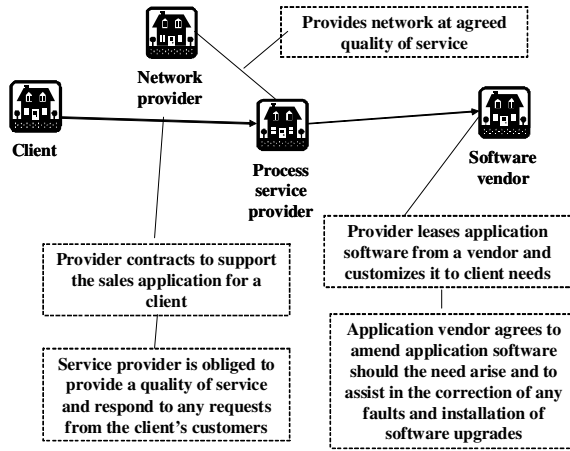


Fig. 3. An outsourcing business arrangement

5 The Business Activity Model

The business activity models are based on a conceptual model for collaborative systems (Hawryszkiewicz, 2005). The main concepts are activity, role, participant, and artefact. Figure 4 illustrates one instance of such model for managing outsourced projects. Figure 4 illustrates the most fundamental parts of a conceptual model of collaboration with more details found in (Hawryszkiewicz, 2005). The concepts used here are activities, roles, artifacts and participants. In Figure 4 there are four activities shown as clouded shapes. Figure 2 is an example of a business activity diagram showing typical activities in an outsourcing arrangement. The diagram also shows a description for each activity. For example:

- “Receive service report” and “Sales recording” are both operational with a task focus,
- “Resolving a service report” can be classified as at the operational management level, often of a collaborative nature.
- “Arrange program change” has a mix of different work kinds and hence should probably be decomposed into two activities, one to decide what change is needed and the other to coordinate the change implementation.

There seven roles shown by as black dots and five artifacts shown by the disk shapes. Any number of participants (not shown in this simplified diagram) can be assigned to each role. The model shows that the client and marketing manager interact in activity 'analysis of marketing needs' to develop a market report. The additional detail includes various discussion or interaction artifacts and ways to initiate events in one activity that are passed to roles in other activities.

The model semantics support dynamic changes to the model and the special characteristics of CAS as:

- They allow activities to be reorganized through changes to roles, and artifacts,
- New activities can be set up and linked to existing activities through roles and artifacts,
- The activities are loosely coupled through their roles,
- New connections can be organized through events or shared discussions,
- Higher level activities can be created to aggregate the activities of existing activities.

Transient activities may be constructed to resolve an urgent issue. Thus a virtual team is created to solve an issue. The project leader 'o1' becomes the coordinator of this team and members are assigned from the provider and vendor teams.

The next step is now to create the work network by looking at each activity and matching it to a work pattern. These work patterns are then combined into the one work diagram.

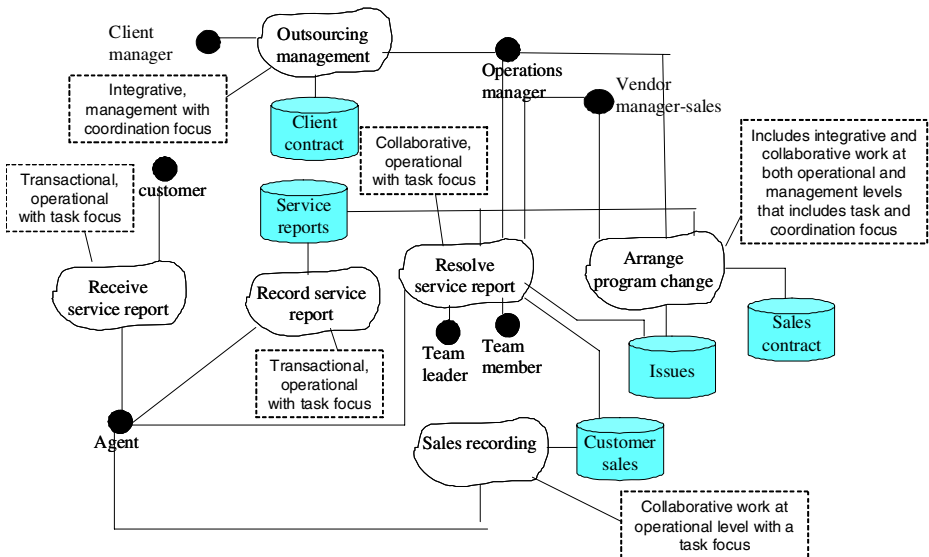


Fig. 4. A business activity diagram also showing the activity descriptions

6 Social Network Models - Extending Network Diagrams to Collaboration Graphs

These have been widely used to model relationships between people. These have been extended in a variety of ways to suit different purposes. Business collaboration requires a clearer definition of what the people do and how they should collaborate in their work. At the same time the chosen structures must naturally support the social acceptance of any new design.

Any such model must specify people roles in the business and the interactions between the roles. Collaboration is often seen in the small and the large. Collaboration in the small mainly focuses on teams or small communities of practice. Communication in the large focuses on cross unit or cross organizational collaboration. Any modelling must cover both of these. It must also take socio-technical issues into consideration in choosing relationships and social structures.

There are a number of commonly found role structures that can be used as guidelines in design. They can then be used as standard patterns used to define collaboration networks in design.

6.1 Coordination Pattern

Here as shown in Figure 5, the coordinator ensures that a number of teams work towards the same goal. Figure 5 shows three teams each organized as a task team. They both look at the outputs of each team and the progress of the team in creating the outputs. The structure shown in Figure 5 can also apply to leadership roles where the leader is required to manage a number of teams. Each team here can be modeled as a separate activity, usually task oriented work. The coordinator and team leaders together plan the project and agree on resources and completion times. The four roles together can be seen as a separate mostly collaborative activity.

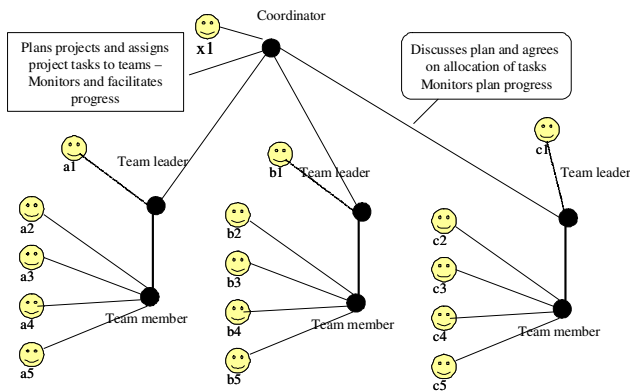


Fig. 5. Coordination Role

6.2 Creating Enterprise Collaboration Networks

The kind of work in general identifies the collaboration pattern for the activity, the management level identifies the kind of communication and the activity focus defines the kind of artifact produced. There are other parameters that play a role in defining business requirements. These apart from the size of the collaborating group, are its geographic distribution, the complexity of their goal, and any time constraints placed on getting some outcome. It also depends on the type of work supported.

The social component focuses on setting up community spaces for the actors in the different innovation activities. The goal is to identify the kind of relationships that exist between work roles in the activities and map them onto activities. The social relationships are based on common communication patterns found in business. This can include brokering, leadership, facilitation and so on.

Ultimately a collaborative structure is made up of a set of interleaving small scale collaborations each with a particular goal and each following a different collaboration pattern.

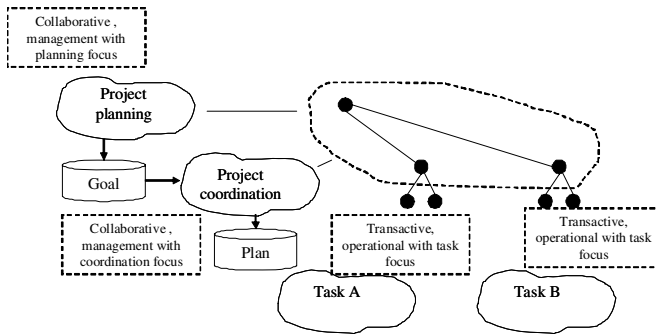


Fig. 6. Integrating patterns

7 The Collaboration Graph for an Outsourcing Example

The work network is now constructed using the kind of approach illustrated earlier in Figure 6.

We look at the activity description and match a social pattern to the activity. A different pattern is constructed for each of the teams, which are primarily collaborative at the operational management level and focus on task execution. Figure 6 shows:

- The roles and role participants. The roles are shown as dots whereas the participants are shown as faces. For example 'a1' is the team leader in the provider team,
- The collaboration within activities shown by the circles,
- The role responsibilities shown by the dotted boxes linked to the roles,
- The interactions between the roles shown in dotted boxes linked by dotted lines to the interaction.

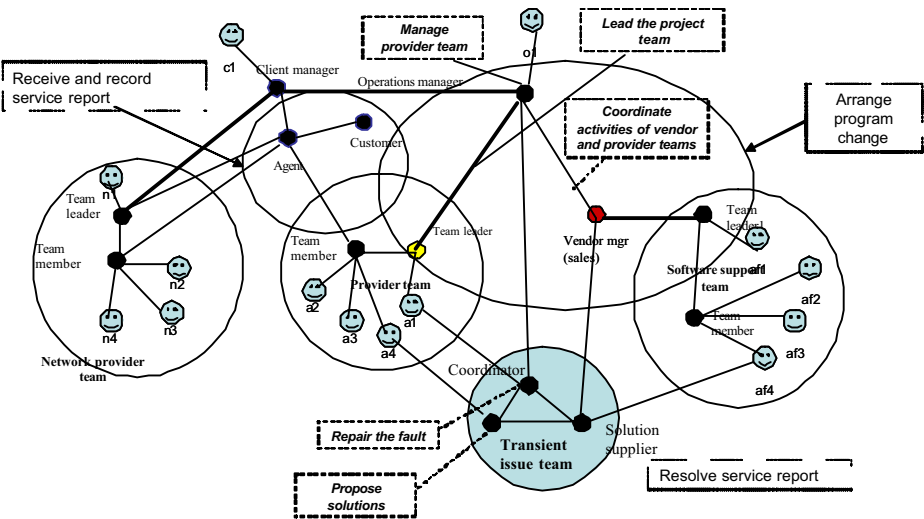


Fig. 7. The Social Network Diagram for process outsourcing

Here we show the responsibilities and interactions on the diagram. These tend to be brief for illustrative purposes. Actual documentation is more complete and can be provided separate from the diagram.

8 Mapping to an Implementation

The simplest mapping is for the outsourcing example is shown in Figure 7.

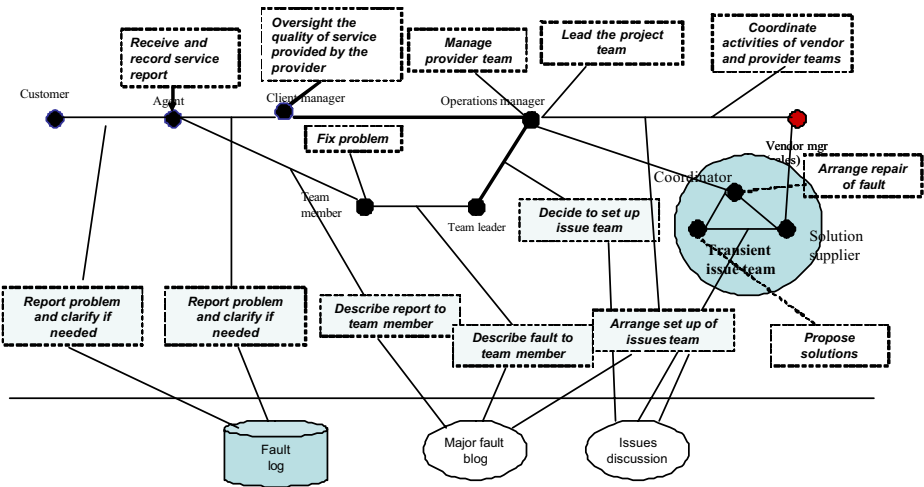


Fig. 8. Choosing Technologies

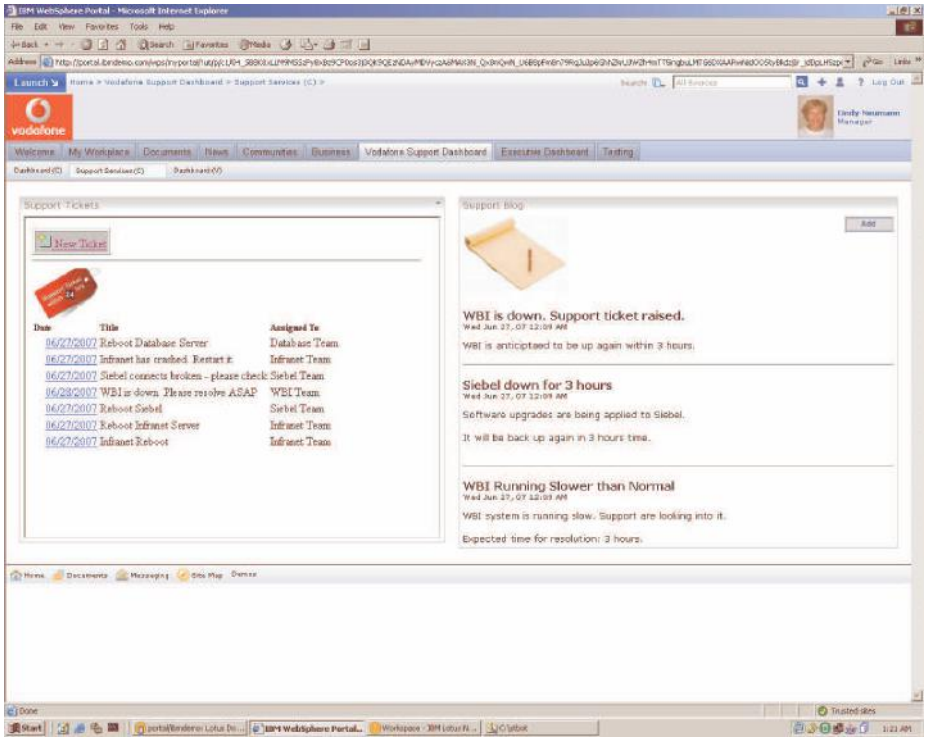


Fig. 9. A role based interface

Implementation then focuses on selecting each role and including all the collaborative relationships and business objects from the roles activities into the workspace. Figure 10 illustrates an example using websphere.

Other implementation can also be based on lightweight platforms (Hawryszkiewicz, 2007) that enable users to themselves customize the workspace to their needs.

9 Summary

The paper raised the importance of including collaboration between users at all levels of design of complex systems. Complex systems evolve over time and such evolution focuses on changing collaborative needs of system users. The paper proposed a blueprint for this purpose. The blueprint is made up of three major parts, namely, business activities, collaboration graph and knowledge management. It then illustrated the modelling methods using process outsourcing as an example.

References

1. Davenport, T.: Thinking for a Living. Harvard Business School Press (2005)
2. Gregor, S., Jones, D.: The Anatomy of a Design Theory. Journal of the Association of Computing Machinery 8(5), 312–335 (2007)

3. Hawryszkiewicz, I.T.: A Metamodel for Modeling Collaborative Systems. *Journal of Computer Information Systems* XLV,(3), 63–72 (Spring 2005)
4. Hawryszkiewicz, I.T.: Lightweight Technologies for Knowledge Based Collaborative Applications. In: *Proceedings of the IEEE CEC/EEE 2007 Conference on E-Commerce Technology*, Tokyo, July 2007, pp. 255–264 (2007)
5. Holland, J.: *Hidden order: How adaption builds complexity*. Cambridge Perseus Books (1995)
6. Kodama, M.: New knowledge creation through leadership-based strategic community – a case of new product development in IT and multimedia business fields. *Technovation* 25, 895–908 (2005)
7. Kovacs, A.I., Ueno, H.: Towards Complex Adaptive Information Systems. In: *Proceedings of the 2nd International Conference on Information Technology and Application* (2004)
8. McAfee, A.P.: *Enterprise 2.0: The Dawn of Emergent Collaboration*. MIT Sloan Management Review, pp. 21–28 (Spring 2006)
9. Morgan, G.: *Images of Organization*. SAGE Publications, Beverly Hills (1986)
10. Merali, Y., McKelvey, B.: Using Complexity Science to effect a paradigm shift in Information systems for the 21st century. *Journal of Information Technology* 21, 211–215 (2006)

Designing Modular Architectures for Cross-Organizational Electronic Interaction

Christoph Schroth^{1,2}, Beat Schmid¹, and Willy Müller³

¹ University of St. Gallen, MCM Institute, 9000 St. Gallen, Switzerland

² SAP Resesarch, Blumenbergplatz 9, 9000 St. Gallen, Switzerland

³ Federal Strategy Unit for IT (FSUIT), 3003 Bern, Switzerland
christoph.schroth@cdtm.de, beat.schmid@unisg.ch,
willy.mueller@isb.admin.ch

Abstract. Information and Communication Technologies (ICT) have paved the way for a new era of industrialization. Services, which can be consumed via Web browsers and allow for the location-independent trade with products and services of all kinds already, are about to be transformed into Web services, i.e. modules, which are machine-processible and can be integrated into globally dispersed, dynamic value chains. The brokering between service consumers and providers requires the design of novel electronic business media which follow and augment the vision of Service-oriented Architectures (SOAs). In this paper, we propose a modular architecture framework for the organization and implementation of such media.

Keywords: Modular Architectures, Cross-Organizational Electronic Interaction.

1 Introduction

Nowadays companies have to manage a rising number of dynamic inter-organizational business relationships, calling for means that allow for their efficient and effective management. Gartner Research emphasizes the increased relevance of information technology (IT) in this context: “We expect that by 2011, midsize- and large companies will have at least doubled the number of multienterprise integration and interoperability projects they're managing and will be spending at least 50% more on B2B projects, compared with 2006. We also believe that, from 2008 to 2013, multienterprise traffic will at least triple.” [2, p.2].

Since the 1960s, IT has been employed to increase the efficiency of cross-organizational interaction by reducing processing errors as well as the time required for data retrieval and translation. In the 1990s, first Internet-based technologies emerged which were not limited to mere data exchange, but also allowed to realize more complex organizational models. During the past years, Service-Oriented Architectures (SOAs) have become an acknowledged general architectural style underlying the implementation of cross-organizational electronic interaction. The widely accepted normative OASIS Reference Model for SOA defines SOA as “...a paradigm for organizing and utilizing distributed capabilities that may be under the control of

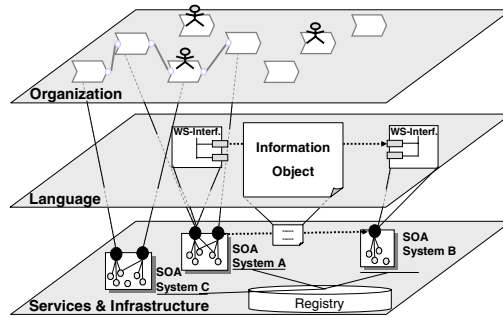


Fig. 1. Basic Service-Oriented Architecture (SOA)

different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable pre-conditions and expectations”[3].

Figure 1 illustrates the fundamental concept of SOA [6]. Existing technical standards such as the Web services stack today allow for the actual implementation of SOAs that span across company boundaries. However, as analyses have shown [6], considerable “silo walls” still exist on all three above mentioned layers, preventing from operational agility:

On the first, technical level, *agents* and the services they provide have to be connected physically in order to allow for their interaction. Existing platforms and standards for the implementation of cross-organizational business relationships can mostly be considered as proprietary island solutions [6, 2]. In this paper, we argue for an augmented version of the Event-Bus Schweiz [4] standard which allows for the setup of a federated event bus infrastructure that shall act similar to a cross-organizational operating system. In accordance with the St. Gallen Media Reference Model (MRM) [5], we call this layer the **C-component** of the medium (C for channel system).

The second “language” level defines the types of the objects of interaction, i.e. the objects, on which the agents act, and which they exchange. Here, a plethora of different, industry-specific and monolithic standards today exist which enable and at the same time often prevent from cross-domain interoperability (e.g., CIDX, HL7, PIDX, RosettaNet business documents, SWIFT, etc.). Today, experts are forced to understand every syntactic and semantic detail of proprietary application interfaces in order to interconnect them. A novel approach is required which provides common design rules for this layer and also proposes a library of modular semantic building blocks that act as common basis for modelling different information objects [7]. We call this layer of the medium its **L-component** (L for language, or logic).

On an organizational level, the types of the interacting agents have to be defined explicitly through the introduction of role descriptions. Also, the procedures of interaction must be defined, be it in a declarative way as rules, or procedurally as traditional processes. Existing methods for modeling cross-organizational interoperation pre-dominantly follow a process-oriented approach. This document argues for a novel

modular method that considers both structural as well as process-oriented organization [6]. We call this part of the definition of the medium its **O-component** (the organizational component).

We refer to the combination of the three components as *medium* which enables the interaction of agents [5]. The following chapter will outline the central design guidelines proposed for the organization and implementation of modular [1] cross-organizational electronic interaction.

2 Architecture Framework for Modular Cross-Organizational Interaction

2.1 C-Component

The C-Component of our proposed architecture framework builds on the two complementary architectural styles of Service-Oriented Architectures (SOAs) and Event-Driven Architectures (EDAs). In specific, it builds on and augments the Event-Bus Schweiz (EBS) specification [4]. The EBS standard establishes a set of design guidelines for building a federated system of numerous event buses (referred to as EBS sub-buses; “Teilbus” in German) which allow for fulfilling heterogeneous, individual requirements and still enable cross-bus interoperability. Figure 2 illustrates the most central components of our extended specification: Rather than implementing the interaction between a set of agents based on one electronic medium, interaction scenarios are decomposed into so called **interaction modules (IAMs)**.¹ For each of those modules, a sub-bus is realized (in this example, a first module IAM0 comprises medium M1 which enables the communicative exchange of agents 1, 2, and 3) which implements a number of services: The *contract structure service* implements the structural organization within each bus medium. It specifies the agents connected to the bus, their roles, and the tasks they are authorized to perform within their respective IAM. The *task structure service* implements the process-oriented organization established within the respective IAM. For each of the allowed tasks, it documents precedence relationships to other tasks. The *object catalogue service* specifies all the information object schemata which may be exchanged via the bus medium. *Operational services* (e.g. encryption, decryption, routing) assume operating system functionality and are well described in [4]. The recursivity inherent in this design allows for information hiding where required and thus supports decoupling: Agent 1, for example, may assume a defined role within IAM0, while it encapsulates the interaction of a number of sub-ordinate agents (A1.1, A1.2, A1.3) who interact via a hidden medium. As long as the general design rules (based on the above outlined services) are considered when designing a modular system of such sub-buses, cross-bus interoperability and efficient redesign is ensured: In case a service provided by a so far hidden agent (e.g., A1.1.1) shall be made available to a greater number of agents, for example, all design information required for its consumption can be “propagated upwards” through the design hierarchy (by means of updating the above mentioned services) [6].

¹ In section 2.3, our method for identifying these interaction modules will be explained.

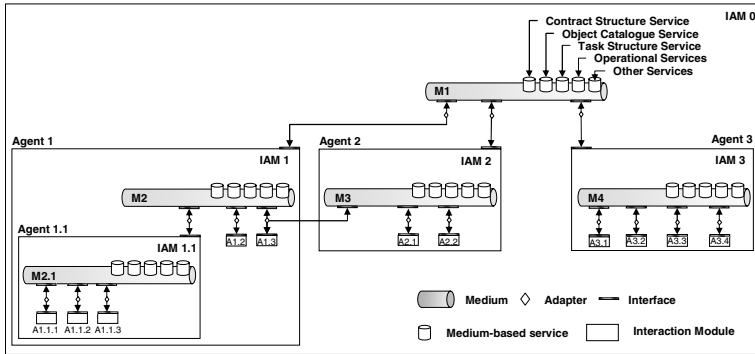


Fig. 2. Modular system of autonomous, but interoperable event-buses

2.2 L-Component

Our architecture framework proposes a modular, core-component-based modeling approach which augments evolving standards such as the OASIS Universal Business Language (UBL), the UN/CEFACT Core Component Technical Specification (CTS), and, on a technical level, the W3C XML schema (see Figure 3). Four abstract entities constitute the nucleus of our information object modeling approach: First, *generic core components* act as reusable, modular building blocks for the design and assembly of comprehensive *generic business documents*. The CTS methodology proposes the four core component types Core Data Type (CDT), Basic Core Component (BCC), Association Core Component (ASCC), and Aggregate Core Component (ACC). Generic (context-neutral) document descriptions (see the rectangle on the left in Figure 1) encapsulate the organization of whole documents such as order or invoice documents. They can be compared with classes in the software programming context as they can be instantiated several times in different contexts. The instantiations of generic business documents are referred to as *specific business documents*. Such specific business documents are constituted of *specific core components*, i.e. the context-specific instantiations of their generic counterparts, the generic core components. The mechanism by which specific documents and core components are derived corresponds to the mechanism of “*restriction inheritance*”. Only those information object constituents are selected that are relevant in a given context (see the three context-specific instantiations on the right side of the figure). Our framework augments the above mentioned standards as it provides a guided procedure for the graphical modeling of unstructured data and its subsequent transformation into standard-compliant data components, as it introduces a comprehensive methodology for the incorporation and management of contextual information, and as it proposes an XML schema-based representation of generic business documents (including context parameters) [7].

2.3 O-Component

To achieve a modular organization of electronic interaction, an interaction scenario first of all has to be decomposed into its constituent, fine-granular tasks (business activities, performed by agents, defined as operations related to specific information objects).

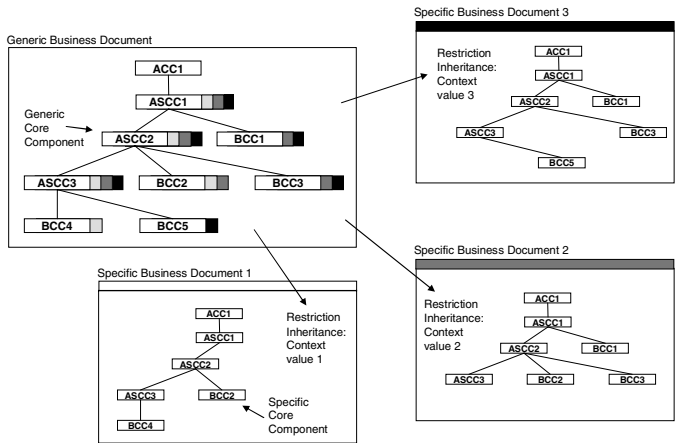


Fig. 3. Modular design of information objects

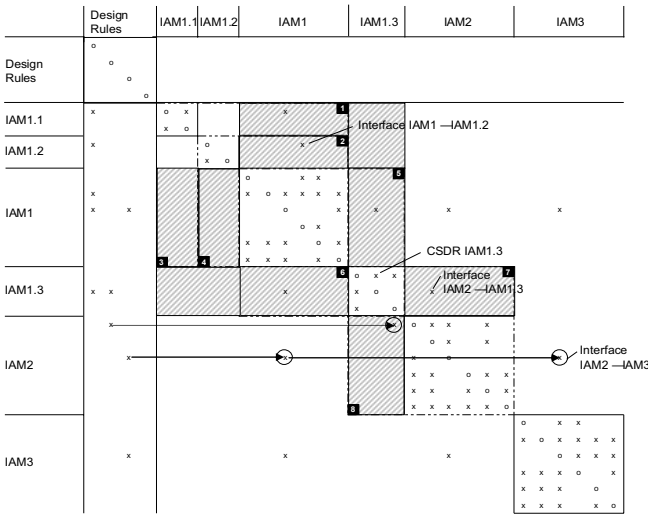


Fig. 4. Service-oriented modeling with task structure matrices [6]

The identified tasks (lowest level of organizational abstraction) are then assigned to both the x and the y axis of a task structure matrix [1, 6] (see Figure 4). In case task i precedes task j, a mark (x) is put in column i and row j of the matrix to document precedence relationships between the various tasks. From the marks, we can identify one-way and two-way interaction patterns (IAPs) which represent the second level of organizational abstraction. IAPs encompass two tasks and feature parameters for detailed behavioral modeling [6]. The resulting fields within the matrix which feature a high amount of marks mean highly interdependent groups of activities and are the

basis for the specification of the afore mentioned interaction modules (IAMs) that reside at the third level of abstraction. As few as possible interdependencies shall exist between the tasks comprised by different IAMs (indicated by off-diagonal xs which are not included in one of the IAMs). These interdependencies either need to be made explicit and become the basis for the development of interfaces (relying on context-specific, descriptive design rules (CSDR) or can be removed through the definition of prescriptive design rules. The organizational design is completed by defining a design hierarchy diagram that clearly specifies the nested hierarchy and the inheritance relationships between the modules (Figure 2 corresponds to Figure 4).

3 Conclusion

In this work, we have addressed the need for a novel architecture framework that provides IT architects fundamental design rules for the organization and implementation of cross-organizational electronic interaction. The framework builds on the design principle of modularity to increase efficiency, flexibility, extensibility, to reduce design and management complexity, to account for uncertainty and finally to enable a decentralized and collaborative evolution of business media for electronic, cross-organizational interaction.

References

1. Baldwin, C.Y., Clark, K.B.: Design Rules: The Power of Modularity, vol. 1. MIT Press, Cambridge (1999)
2. Lheureux, B.J., Malinverno, P.: Market Update for Integration Service Providers. Gartner Research Paper (2008)
3. Mackenzie, M., et al.: OASIS - Reference Model for Service Oriented Architecture 1.0: OASIS (2007)
4. Müller, W.: Event Bus Schweiz. Konzept und Architektur, Version 1.5. Bern, Switzerland, Informatikstrategieorgan Bund, ISB (2007)
5. Schmid, B.F., Lindemann, M.: Elements of a Reference Model for Electronic Markets. In: Thirty-First Annual Hawaii International Conference on System Sciences, vol. 4. IEEE Computer Society, Los Alamitos (1998)
6. Schroth, C., Schmid, B.F.: Reference Architecture for Cross-Company Electronic Collaboration. *International Journal of e-Collaboration* 5(2), 75–91 (2009)
7. Schroth, C., Pemproad, G., Janner, T.: CCTS-based Business Information Modelling for Increasing Cross-Organizational Interoperability. In: Gonçalves, R.J., Müller, J., Mertins, K., Zelm, M. (eds.) *Enterprise Interoperability II. New Challenges and Approaches*, Springer, Heidelberg (2007)

Modelling the Bullwhip Effect Dampening Practices in a Limited Capacity Production Network

Elena Ciancimino and Salvatore Cannella

University of Palermo, Faculty of Engineering
viale delle Scienze, Parco d'Orleans,
90128 Palermo, Italy
{eciancimino, cannella}@unipa.it

Abstract. This work infers on the conjoint adoption of collaboration practices and replenishment rules as bullwhip dampening techniques in a limited capacity production network. Continuous time differential equation methodology is adopted to model three supply chain configurations. Results show that the *conditio sine qua non* for long-term capacity strategy is the management and control of information asynchronies, provided by collaboration practices and ad-hoc decision policies. Furthermore the study reveals the phenomenon of bullwhip rough dampening.

Keywords: multi-echelon, business network, supply chain management, capacity constraints, bullwhip effect, false demand, order policy, periodic review, order-up-to, APIOBPCS, smoothing replenishment, information sharing, EPOS, VMI, supply chain metrics, customer service level.

1 Bullwhip and Production Capacity: An Overview

Modern supply chain management starts with the premise that supply chain members are primarily concerned with optimising their own objectives and this self-serving focus often results in poor performance [1]. A key example of such inefficiency is the bullwhip effect, a global time-varying phenomenon referring to the amplification of orders as they pass up the supply chain from marketplace to upstream suppliers.

The research related to the problem of amplifying signals in multi-echelon production and distribution systems dates back to the first half of the 20th century [2]. The phenomenon was recognised as early as 1919 in the supply chain of Procter and Gamble. In 1924 Mitchell [3] reported about the *false demand*, while in 1958 Forrester [4] addressed the problem of the demand waveforms propagating along the chain and the consequent distortive effects on decision making, naming it *demand amplification*. Lee, Padmanabhan and Whang [5] and [6] published in 1997 two of the most popular papers based on a case study, identifying four causes of the phenomenon and coining the term *bullwhip effect*.

In 2008 Disney and Lambrecht [1] published the first monographic study entirely dedicated to bullwhip effect. In one hundred of history of Operations Management several academics and practitioners ([7], [8], [9], [10], [11], [12], [13], [14]) have been fascinated from the demand amplification enigma.

The investigation on the phenomenon has passed through diverse phases, from empirical and ad hoc studies on bullwhip causes to mathematical approaches to infer on demand amplification solutions. In the new millennium the bullwhip issue entered the *avoidance phase* [2], and the scientific discussion converged toward the efficacy of solution techniques and practices.

In order to infer on the solutions' performance, in this phase researchers are focusing on the impact of avoidance techniques on models that aim at reflecting real business world conditions. One the relevant features of the global enterprise business network is the constrained capacity of production plants and distribution centres. Long-term capacity decisions are among the most fundamental of all that plant managers are called on to make [15]. In the operational layer, the variation of long-term production capacity weighs on job sequencing, resource re-allocation, production lead time, and on the related costs (set-up, stock holding, supplier unstructured contracts, overtime). *The variation of capacity production and bullwhip effect are linked by their own nature*: the amplification of variance of order rates in multi-echelon systems could lead to saturation of the network production capacity.

The paper is organised as follows. In Section 2 the adopted demand amplification phenomenon solutions are illustrated. Section 3 describes the adopted methodology. The mathematical formalism is presented in section 4. Performance metrics, experimental sets and data analysis are reported in section 5. Section 6 provides the conclusions.

2 Bullwhip Solution Approaches: Collaboration Practices and Smoothing Replenishment Rules

Many authors have described and classified the solving approaches to the demand amplification problem and supply chain instability ([16], [17], [18], [13], [2], [19], [20]). In 1993 Van Ackere et al. [21] provided a useful framework to classify the countermeasures that can be taken in any supply chain to dampen or avoid the bullwhip effect [13]. They distinguish three different solving approaches: (1) redesigning the physical process (lead time reduction, eliminating a channel in the supply chain); (2) redesigning the information channel (sharing real-time point-of-sales information, sales forecasts, inventory order policies and inventory reports); (3) redesigning the decision process (ad-hoc replenishment rules). In this paper the second and the third approaches are adopted in a constrained capacity production network.

The redesigning the information channel solving approach is realised through the adoption of supply chain collaboration practices. Over the last decade the operation management community has focused on the evolution of hi-tech applications and the related integrated forms of trading partner coordination mechanisms. As reported by Disney and Towill [22], Vendor managed inventory (VMI), synchronised consumer response (SCR), continuous replenishment (CR), efficient consumer, response (ECR) [23], rapid replenishment (RR), collaborative planning, forecasting and replenishment (CPFR) [24] and centralised inventory management (CIM) [5], refer to supply chain strategies where the vendor or supplier is given the responsibility of managing the customers stock. The practice appellatives depend on sector application, ownership issues and scope of implementation. However, in essence, they are all specific applications of integrated multi-echelon management.

The essence of collaboration practices consists in sharing of operational production network information: real-time point-of-sales, sales forecasts, inventory order policies and inventory reports to support multi-tiers.

The solving approach consisting in redesigning the decision process is commonly realised through the adoption of smoothing replenishment rules. This technique permits to decrease the tiers' lot size in presence of marketplace demand information distortion. The most notorious family of smoothing decision rule belongs to the Inventory and Order Based Production Control System, known as IOBPCS family [7], which consists of a range of Production and Inventory Control systems with five main components: a forecasting mechanism, a set of parameters and time values, an inventory feedback loop, a work in progress feedback loop and a target net stock setting [25]. In 1994 John et al. [26] coined the term Automatic Pipeline Inventory and Order Based Production Control System (APIOBPCS), which also takes into account Work In Progress (WIP), comparing actual levels with a target value.

APIOBPCS is a base-stock (R, S) periodic review/order-up-to with proportional controllers. The proportional controller in an order policy is the smoothing term of the discrepancy between actual and target levels of net stock and pipeline stock.

The aforementioned two solving approaches are investigated in the following supply chain configurations.

(1) Traditional linked production network (TL). Each echelon only receives information on local stock, local work in progress levels, and local sales. The retailer forecasts customer demand on the basis of market time series and the remaining trading partners only take into account for their replenishment downstream incoming orders.

(2) Exchange Point of Sales supply chain (EPOS). All echelons base their inventory policy on local stock, local work in progress levels, local sales, downstream incoming orders and the actual marketplace demand.

(3) Vendor managed inventory supply chain (VMI). The order policy is supported by real-time databases that integrate information on local stock, local work in progress levels, local sales, downstream incoming orders, actual marketplace demand, inventory information and work in progress data incoming from the downstream trading partners.

3 Methodology: Continuous Time Domain

The operation management literature is rich in classifications of methodologies used to investigate on supply chain performance and on the bullwhip phenomenon.

One of the classifications for demand amplification modelling has been discussed by Holweg and Disney [2]. They recognised three distinct and methodologically independent research domains: the discrete time approach, the continuous time approach and the control theory approach. The authors affirm that Herbert Simon [27] and Jay Forrester [4] laid the foundations to the continuous time domain approach towards the study of supply chain dynamics. The Nobel prize discussed the application of linear deterministic control theory to production control, by using Laplace transform techniques [28], while Forrester adopted mono-step numerical methods (Euler-Cauchy method, Kutta's method) to approximate a solution for the initial-value problem of

nonlinear repeated coupling of first-order differential equation systems. Several mathematical toolboxes designed to solve a broad range of problems or ad-hoc applications, such as Vensim, ithink, DYNAMO and Powersim, are used to approximate the solution of the differential equations.

4 Models: Mathematical Formalism

The mathematical formalism of the production/distribution networks is reported in the followings.

$$W_i(t) = W_i(t-1) + S_{i-1}(t) - S_{i-1}(t-T_p) \quad . \quad (1)$$

$$I_i(t) = I_i(t-1) + S_{i-1}(t-T_p) - S_i(t) \quad . \quad (2)$$

$$B_i(t) = B_i(t-1) + \tilde{R}_{i+1}(t) - S_i(t) \quad . \quad (3)$$

$$S_i(t) = \min\{\tilde{R}_{i+1}(t) + B_i(t-1); I_i(t-1) + S_{i-1}(t-T_p)\} \quad . \quad (4)$$

$$\tilde{R}_i(t) = \min\{R_i(t); cf\} ; R_i(t) \geq 0 \quad . \quad (5)$$

$$\hat{d}_i(t) = \alpha R_{i+1}(t-1) + (1-\alpha)\hat{d}_i(t-1) \quad . \quad (6)$$

$$R_{K+1}(t) = d_{market}(t) \quad . \quad (7)$$

$$TW_i(t) = T_p \hat{d}_i(t) \quad . \quad (8)$$

$$TI_i(t) = T_c \hat{d}_i(t) \quad . \quad (9)$$

$$VirtW_i(t) = \sum_{j=i}^K W_j(t) \quad . \quad (10)$$

$$VirtI_i(t) = \sum_{j=i}^K I_j(t) \quad . \quad (11)$$

$$TVirtW_i(t) = \hat{d}_K(t) \sum_{j=i}^K T_p \quad . \quad (12)$$

$$TVirtI_i(t) = \hat{d}_K(t) \sum_{j=i}^K T_c \quad . \quad (13)$$

$$R_i(t) = \hat{d}_i(t) + \frac{1}{T_w}(TW_i(t) - W_i(t)) + \frac{1}{T_y}(TI_i(t) - I_i(t)) \quad (14)$$

$$R_i(t) = \hat{d}_K(t) + \frac{1}{T_w}(TW_i(t) - W_i(t)) + \frac{1}{T_y}(TI_i(t) - I_i(t)) \quad (15)$$

$$R_i(t) = \hat{d}_K(t) + \frac{1}{T_w}(TVirtW_i(t) - VirtW_i(t)) + \frac{1}{T_y}(TVirtI_i(t) - VirtI_i(t)) \quad (16)$$

In table 1 the production network structures are associated with the respective equation system. In table 2 the model nomenclature is presented.

Table 1. Model structures and the respective equations

Structure	Equation
<i>TL</i>	(1), (2), (3), (4), (5), (6), (7), (8), (9), (14)
<i>EPOS</i>	(1), (2), (3), (4), (5), (6), (7), (8), (9), (15)
<i>VMI</i>	(1), (2), (3), (4), (5), (6), (7), (10), (11), (12), (13), (16)

Table 2. Nomenclature

W_i	<i>work in progress</i>	$TVirtW_i$	<i>target virtual work in progress</i>
I_i	<i>inventory of finished materials</i>	$TVirtI_i$	<i>target virtual inventory</i>
S_i	<i>units/orders finally delivered</i>	d	<i>customer demand</i>
$\hat{d}$	<i>customer demand forecast</i>	cf	<i>capacity factor</i>
R_i	<i>replenishment order quantity</i>	α	<i>forecast smoothing factor</i>
$\tilde{R}_i$	<i>capacitated replenishment order</i>	T_p	<i>physical production/distribution lead</i>
B_i	<i>backlog of orders</i>	T_c	<i>cover time for the inventory control</i>
$VirtW_i$	<i>virtual work in progress</i>	T_w	<i>smoothing work in progress parameter</i>
$VirtI_i$	<i>virtual inventory</i>	T_y	<i>smoothing inventory parameter</i>

5 Performance Metrics, Experimental Sets, Data Analysis

The metrics adopted to assess the simulation results are the Order Rate Variance Ratio [13] with the Disney and Towill variation [29], the Average Backlog, the Average Inventory, the Zero Replenishment and the Bullwhip Slope [30].

$$ORVrRatio = \sigma_R^2 / \sigma_d^2 . \quad (17)$$

$$AverageInventory = \frac{1}{T} \sum_{i=0}^T I_i(t) . \quad (18)$$

$$GAI = \frac{1}{T} \sum_{i=0}^T I_i(t) . \quad (19)$$

$$AverageBacklog = \frac{1}{T} \sum_{i=0}^T B_i(t) . \quad (20)$$

$$ZR_i = \sum_{i=0}^T x_i(t); \quad x_i(t) = \begin{cases} 1 & R_i(t) = 0 \\ 0 & R_i(t) \neq 0 \end{cases} . \quad (21)$$

The *Order Rate Variance* (ORVr) Ratio metric (17) is a smart and concise quantification of the order rate instability. The theoretical value of (17) is equal to 1 in case of absolute absence of demand amplification. σ_R^2 is the variance of the order quantity and σ_d^2 stands for the variance of market demand. The higher the value of ORVr Ratio, the greater is the magnitude of demand amplification phenomenon.

A geometric or exponential increase of ORVr Ratio in upstream direction in the supply chain is representative of the transmission of bullwhip effect [19]. The values of ORVr Ratio are interpolated along each chain and the curve slopes are calculated. The *slope* of ORVr Ratio is a single value, which is indicative of the extent of bullwhip effect propagation and inventory instability along a given supply chain structure.

The *Average Inventory* (18) is computed as the mean value of Inventory at echelon i over the simulation time span T . A concise measure of multi-echelons system performance related to stock levels is the *Global Average Inventory* (GAI), computed as the sum of Average Inventory values over the four tiers.

As customer service level measure, Backlog (3) is adopted to quantify the accumulation of unfilled orders. The Backlog is evaluated every single Δt and the time series reproduce the supply chain customer service level history. To associate a customer service level indicator to each supply chain and concisely compare different scenarios, an additional measure is used: the *Average Backlog* (23).

For fixed cycled inventory policies, the circumstance associated to a null argument in a replenishment order quantity is herein defined *Zero-Replenishment* (ZR) phenomenon. The Zero-Replenishment (21) is quantified as the number of times in which tier i does not place any order, while market demand has reached a stable value. ZR is a measure of timely and ponderated reactivity of a tier's operations towards changes in

demand. It provides an assessment of supply chain scalability: the ability of business manufacturing, or technology process, to support sudden increases in demand. A high value of ZR is indicative of an excessive dimensioning of the order lot size. The ZR phenomenon can also be evaluated at system level referring to a *Global Zero-Replenishment* (GZR), computed as the sum of the ZR_i of the single echelons in a given supply chain. Note that ZR shall be analysed conjointly with a customer service level assessment: to affirm that a system is reacting timely and ponderately, a good service level has to be associated to a low value of ZR. The ZR alone cannot be viewed as a stand-alone supply chain performance metric. Apparently a low value of ZR is indicative of optimal operations and lot sizing: this is true only when at the same time the system assures a high customer service level. Otherwise, a poor customer service level associated to a low ZR reflects the exact contrary: poor system reactivity.

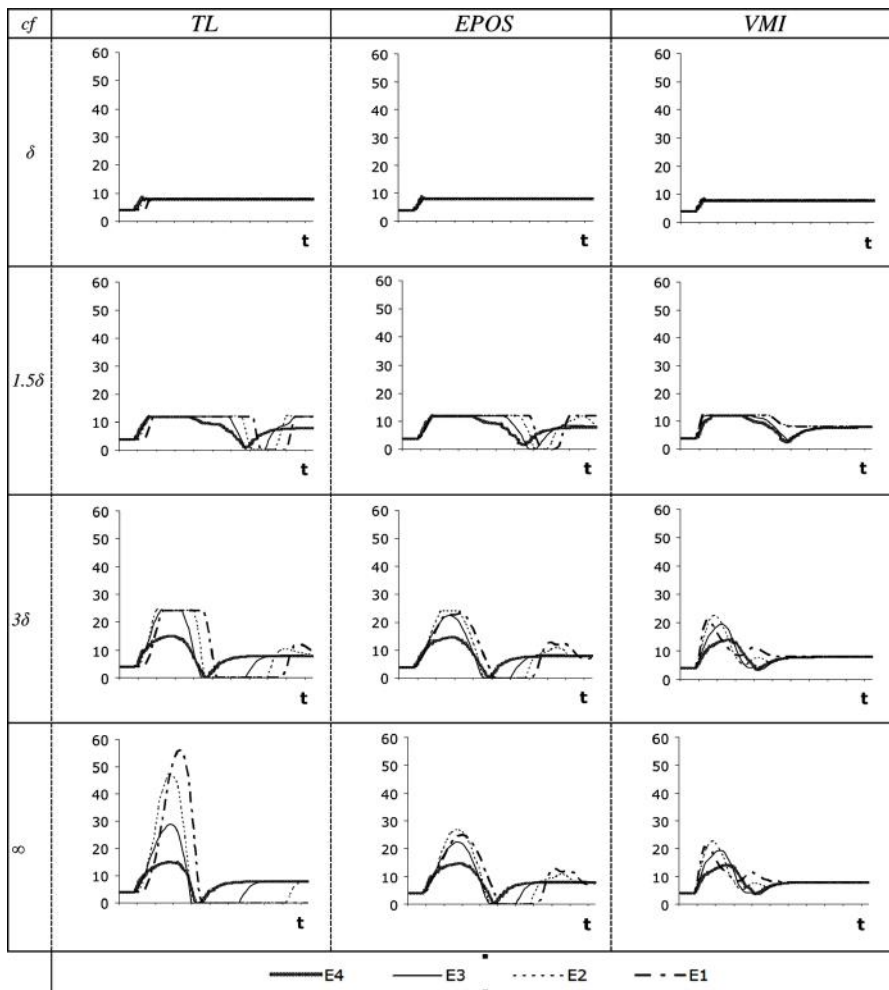


Fig. 1. Order quantities plotted for structure (columns) and for capacity level (rows)

5.1 Experimental Sets and Results

In this study three capacity constraint levels and one unconstrained case are analysed. The limited capacity condition is obtained by modelling a capacity factor *cf*. The capacity factor is a linear function of the final marketplace steady state demand δ . This relation between capacity constraints and customer demand is assumed also in Evans and Naim [31] and Simchi-Levi and Zhao [32]. In this work the levels of capacity factor are $[\delta; 1.5\delta; 3\delta; \infty]$.

Each simulation is run for a total of 52 time units, with order of accuracy equal to $\Delta t=0.25$. The marketplace demand is initialised at 4 units per time unit, until there is a pulse at $t=5$, increasing the demand value up to $\delta=8$ units per time unit. The state values at $t=0$ are the same as Sterman's [10] configuration.

The following values are used for the parameters: forecast smoothing factor $\alpha=0.5$; physical production/distribution lead time $T_p=2$; cover time for the inventory control $T_c=3$; smoothing inventory parameter $T_y=3$; smoothing work in progress parameter $T_w=3$. The smoothing inventory parameter T_y and the smoothing work in progress parameter T_w are chosen on the basis of the empirical formula $T_y=T_w=1+T_p$, [33]. Note that the Deziel and Eilon [34] smoothing parameter configuration is used.

The system is constituted by $K=4$ echelons. For echelon 1, the manufacturer, no replenishment lead time is considered ($T_p=0$).

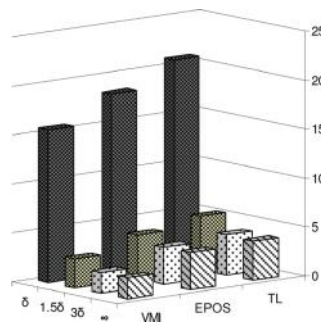
Table 3. Order Rate Variance Ratio and Bullwhip Slope values

<i>Echelon</i>	δ			1.5δ		
	<i>TL</i>	<i>EPOS</i>	<i>VMI</i>	<i>TL</i>	<i>EPOS</i>	<i>VMI</i>
4	1,05	1,05	1,05	7,11	6,65	5,37
3	1,14	1,11	1,02	13,44	10,13	5,76
2	1,23	1,14	1,00	15,69	11,92	5,33
1	0,76	0,59	0,52	8,42	6,98	2,32
<i>slope</i>	0,09	0,04	0,02	4,29	2,64	0,02

<i>Echelon</i>	3δ			∞		
	<i>TL</i>	<i>EPOS</i>	<i>VMI</i>	<i>TL</i>	<i>EPOS</i>	<i>VMI</i>
4	9,78	9,13	5,18	9,98	9,15	5,18
3	45,29	29,69	12,32	52,72	29,70	12,32
2	60,17	42,74	14,35	149,70	46,52	14,35
1	34,60	21,79	4,87	116,12	23,00	4,87
<i>slope</i>	25,19	16,80	4,59	69,86	18,68	4,59

Table 4. Global Zero Replenishment and Global Average Inventory values

	GZR				GAI			
	δ	1.5δ	3δ	∞	δ	1.5δ	3δ	∞
<i>TL</i>	0	22	52	74	57	109	196	57
<i>EPOS</i>	0	14	33	34	61	85	85	61
<i>VMI</i>	0	0	0	0	66	73	73	66


Fig. 2. Values of Average Backlog

6 Conclusions

This work inferred on the conjoint adoption of collaboration practices and replenishment rules as bullwhip dampening techniques in a limited capacity production network. Continuous time differential equations were adopted to model three supply chain configurations. The bullwhip metrics used were Order Rate Variance Ratio with the Disney and Towill variation, Average Backlog, Average Inventory, Zero Replenishment and Bullwhip Slope. The main results of this study are summarised in the followings.

(I) *The status of perpetuus stockholding.* From an initial analysis of simulation outcomes it seems that low values of capacity factor would improve supply chain performance. In particular, each supply chain configuration (TL, EPOS and VMI) characterised by a capacity factor equal to the steady state demand exhibits optimum values of Order rate Variance Ratio, Global Inventory Average and Global Zero replenishment. The myopic analysis of this metrics yields an unreal interpretation of supply chain performance. An analysis of Backlog clarifies and completes the inference: in the $cf=\delta$ set capacity saturation impedes to recover the accumulation of previous unfulfilled orders and causes a high loss of customer service level. In this experimental set, the result shows a total absence of bullwhip phenomenon provided by the limitations of order rate due to the production/distribution capacity saturation. On the other hand the consequence of

the increment in marketplace demand concretises in another dimension of the supply chain system: a status of *perpetuus stockholding*.

(II) *The bullwhip rough dampening and the noxious residual of information distortion.* Monitoring the backlog in each experimental set for the TL structure suggests that increments of production capacity do not necessarily cause an improvement in customer service. On the contrary, the other metrics show an increment in demand amplification and inventory instability for progressive level of *cf*. These results might suggest that the capacity constraint provides a general improvement of process performance within the multi-echelon system for a step input in demand, in terms of demand amplification and supply chain stability. The result might induce to think that the capacity constraint acts like a bullwhip reducer, but this interpretation is actually a *pink elephant* due to a limitation of the models herein presented and of the performance assessment system. The bullwhip reduction associated to the constrained capacity is an apparent improvement of the performance of the supply chain. *The order quantity is lessened by a saturation of the available production/distribution capacity, but the noxious residual of information distortion persists in traditional linked supply chains.* A worth-noting point is that this *rogue dampening* of the bullwhip effect does produce a *double risk*, as it can lead to satisfy at a higher cost the Mitchell's *false demand*. In other words, a *production capacity extension strategy* is not to be considered a bullwhip dampening technique.

(III) *The conditio sine qua non for long-term capacity management strategy.* The conjoint adoption of smoothing replenishment rules and information sharing enables the supply chains to dampen the demand amplification and avoid the deleterious consequences of capacity saturation. The collaboration practices provide inventory stability, ponderated order rate based on actual marketplace demand and on time purchasing orders, and enables the business network to compete in the World Class Customer Care Era. The *conditio sine qua non* for long-term capacity strategy is the management and control of information asynchronies.

References

1. Disney, S.M., Lambrecht, M.R.: On Replenishment Rules, Forecasting, and the Bullwhip Effect in Supply Chains. *Foundations and Trends in Technology, Information and Operations Management* 2, 1–80 (2008)
2. Holweg, M., Disney, S.M.: The Evolving Frontiers of the Bullwhip Problem. In: *EurOMA: Operations and Global Competitiveness*, Budapest, pp. 777–716 (2005)
3. Mitchell, T.: Competitive illusion as a cause of business cycles. *Quarterly Journal of Economics* 38, 631–652 (1923)
4. Forrester, J.: Industrial dynamics: a major break though for decision-makers. *Harvard Business Review* 36, 37–66 (1958)
5. Lee, H.L., Padmanabhan, V., Whang, S.: Information distortion in a supply chain: the bullwhip effect. *Management Science* 43, 546–558 (1997)
6. Lee, H.L., Padmanabhan, V., Whang, S.: The Bullwhip effect in supply chains. *Sloan Management Review* 38, 93–102 (1997)
7. Towill, D.R.: Dynamic analysis of an inventory and order based production control system. *International Journal of Production Research* 20, 369–383 (1982)

8. Houlihan, J.B.: International supply chain management. *International Journal of Physical Distribution and Materials Management* 17, 51–66 (1987)
9. Serman, J.D.: Modelling managerial behaviour: misperceptions of feedback in a dynamic decision-making experiment. *Management Science* 35, 321–339 (1989)
10. Burbidge, J.L.: Period batch control (PBC) with GT – the way forward from MRP. In: *BPICS Annual Conference*. Birmingham (1991)
11. Wikner, J., Towill, D.R., Naim, M.M.: Smoothing supply chain dynamics. *International Journal of Production Economics* 22, 231–248 (1991)
12. Chen, F., Drezner, Z., Ryan, J.K., Simchi-Levi, D.: Quantifying the bullwhip effect in a simple Supply Chain: the impact of forecasting, lead-times and information. *Management Science* 46, 436–443 (2000)
13. Dejonckheere, J., Disney, S.M., Lambrecht, M.R., Towill, D.R.: The impact of information enrichment on the bullwhip effect in Supply Chains: A control engineering perspective. *European Journal of Operational Research* 153, 727–750 (2004)
14. Warburton, R.D.H.: An Analytical Investigation of the Bullwhip Effect. *Production and Operations Management* 13, 150–160 (2004)
15. Grubbström, R.W., Wang, Z.: A stochastic model of multi-level/multi-stage capacity-constrained production–inventory systems. *International Journal of Production Economics* 81–82, 483–494 (2003)
16. Riddalls, C.E., Bennett, S., Tipi, N.S.: Modelling the dynamics of supply chains. *International Journal of Systems Science* 31, 969–976 (2000)
17. Disney, S.M., Naim, M.M., Potter, A.T.: Assessing the impact of e-business on supply chain dynamics. *The International Journal of Production Economics* 89, 109–118 (2004)
18. Kleijnen, J.P.C., Smits, M.T.: Performance metrics in supply chain management. *The Journal of Operational Research Society* 54, 507–514 (2003)
19. Geary, S., Disney, S.M., Towill, D.R.: On bullwhip in supply chains - historical review, present practice and expected future impact. *International Journal of Production Economics* 101, 2–18 (2006)
20. Towill, D.R., Zhou, L., Disney, S.M.: Reducing the bullwhip effect: Looking through the appropriate lens. *International Journal of Production Economics* 108, 444–453 (2007)
21. van Ackere, A., Larsen, E.R., Morecroft, J.D.W.: Systems thinking and business process redesign: An application to the beer game. *European Management Journal* 11, 412–423 (1993)
22. Disney, S.M., Towill, D.R.: On the bullwhip and inventory variance produced by an ordering policy. *Omega, the International Journal of Management Science* 31, 157–167 (2003)
23. Cachon, G., Fisher, M.: Campbell Soup's continuous replenishment program: Evaluation and enhanced inventory decision rules. *Productions and Operations Management* 6, 266–276 (1997)
24. Holmstrom, J., Framling, K., Kaipia, R., Saranen, J.: Collaborative planning forecasting and replenishment: new solutions needed for mass collaboration. *Supply Chain Management: An International Journal* 7, 136–145 (2002)
25. Lalwani, C.S., Disney, S.M., Towill, D.R.: Controllable, Observable and controllable state space representations of a generalized Order-Up-To policy. *International Journal of Production Economics* 101, 173–184 (2006)
26. John, S., Naim, M.M., Towill, D.R.: Dynamic analysis of a WIP compensated decision support system. *International Journal of Management Systems Design* 1, 283–297 (1994)
27. Simon, H.A.: On the application of servomechanism theory to the study of production control. *Econometrica* 20, 247–268 (1952)

28. Axsäter, S.: Control theory concepts in production and inventory control. *International Journal of Systems Science* 16, 161–169 (1985)
29. Disney, S.M., Towill, D.R.: The effect of vendor managed inventory (VMI) dynamics on the Bullwhip Effect in supply chains. *International Journal of Production Economics* 85, 199–215 (2003)
30. Cannella, S., Ciancimino, E.: The APIOBPCS Deziel and Eilon parameter configuration in supply chain under progressive information sharing strategies. In: *Winter Simulation Conference*, Miami (2008)
31. Evans, G.N., Naim, M.M.: The dynamics of capacity constrained supply chains. In: *International System Dynamics Conference*, Stirling, Scotland, pp. 28–35 (1994)
32. Simchi-Levi, D., Zhao, Y.: The value of information sharing in a two-stage supply chain with production capacity constraints. *Naval Research Logistics* 50, 888–916 (2003)
33. Disney, S.M., Towill, D.R.: A methodology for benchmarking replenishment-induced bullwhip. *Supply Chain Management: An International Journal* 11, 160–168 (2006)
34. Deziel, D.P., Eilon, S.: A linear production: inventory control rule. *The Production Engineer* 43, 93–104 (1967)

A Comparative Analysis of Knowledge Management in SMEs

Maria R. Lee¹ and Yi-Chen Lan²

¹ Shih Chien University, Taiwan
maria.lee@mail.usc.edu.tw

² University of Western Sydney
y.lan@uws.edu.au

Abstract. Organizations have long been acknowledged that knowledge management (KM) is an important aspiring tool for gaining competitive advantages and improving performance. However, many small and medium-sized enterprises (SMEs) face the issues of recognition of real benefits, participation of advancement and transformation. Therefore, they are usually encountering ambiguity and uncertainty of adopting and implementing KM. This study is extended to SMEs along with incubated companies and micro-businesses, and conducts a comparative analysis of KM in SMEs in Taiwan and Hong Kong. The research results indicate that a successful KM implementation depends on a harmonious amalgamation of infrastructure and process capabilities, including technology, culture, and organizational structure. This analysis may also help in understanding the impact of knowledge sharing between government and SMEs, and creating new business values for SMEs.

Keywords: Knowledge management, SMEs, Comparative analysis, Taiwan, Hong Kong, Infrastructure capability, Process capability, acquisition, conversion, application, protection.

1 Introduction

Organizations have long been acknowledged that KM is an important aspiring tool for gaining competitive advantages and improving performance [3], [5]. In most large corporations, the development of KM Systems is implemented in a formal approach [6]. This is evidenced by allocating the substantial portion of the corporate Information and Communications Technology (ICT) budget to KM Systems. Most organizations believe that the evolution of ICT will provide the enabling platform and facilitate KM in business operations.

In contrast, the Management Information Systems (MIS) development agenda in most SMEs is neglected to incorporate KM as part of the plan. There are various reasons for such negligence including budget constraints, shortage of dedicated human resources, rapid change of personnel, lack of understanding the processes involvement in KM, lack of realizing the complexity and different types of knowledge, lack of anticipated delivery and recognizing the immediate benefit of implement appropriate KM systems [8]. Recent years, many researchers have been focusing on the

development of practical implementation of KM in SMEs [2], [3], [6], [9]. These guidelines and manuals have helped the SMEs in the KM venture. Nevertheless, there are issues exist at the initial stage where SMEs fail to realize and recognize the potential benefits of KM.

In Asia, Taiwan and Hong Kong had once been called “Asia Four Little Dragons”, which was referring to the leading four economics in Asia. The overall performance of SMEs in Taiwan and Hong Kong is quite similar. For example, the Ratio of SMEs to all enterprises in Taiwan is 97.63% and 77.12% of total employment according to Taiwan Government statistics circa 2007 whereas SMEs represent 98% to all enterprises and 50% of total employment in Hong Kong. Both Taiwan and Hong Kong have built much of their economic success on SMEs. The paper examines the relative performance of KM of SMEs in Taiwan and Hong Kong. It seeks to extract some KM lessons in SMEs for improving efficiency and business performance.

The main objectives of this study are to firstly understand the issues the SMEs have been facing in adopting and incorporating KM as part of their key business competency, and secondly determine the readiness of SMEs in KM implementation. The paper starts with the literature review based on the organizational capabilities and factors in KM by Chan and Chao [2] and Gold et al. [4]. Through the Taiwanese SMEs KM Survey, the authors compare the results with the study in Hong Kong SMEs by Chan and Chao [2]. The results also help understanding the government role in providing necessary assistance to the SMEs for pursuing KM. The paper concludes with a list of topics such as exploiting the latest Information and Communications Technology (ICT) and conversational approach to KM in SMEs for future research development.

2 Knowledge Management

The mission of implementing a successful KM Systems in SMEs requires a balanced combination of support from management, structure of the organization, and enabling technology [2]. Based on Gold et al [4], Chan and Chao adopted and used “infrastructure capability” and “process capability” in their experiment of 68 SMEs in Hong Kong, which have KM initiatives launched in the past few years.

Three factors are discussed in infrastructure capability namely, Technology, Structure, and Culture. In Technology factor, participating SMEs were asked whether their organizations have the appropriate technology platform and enabling technology to facilitate the process activities of KM. In addition, SMEs were asked if their employees are able to participate in collaborative work through the latest ICT. In Structure factor, questions such as whether the organizational structure encourages employees to get involved in knowledge related process activities. For instance, the organization has a transparent reporting mechanism to support employees with communication and engagement channel for knowledge sharing activity. Another example would be an incentive scheme implemented by the organization to encourage knowledge sharing amongst work colleagues. The third factor in infrastructure capability relates to organizational culture. Further to the “hard” approach to knowledge sharing in organizational structure, the organizational culture plays an imperative element as the “soft”

approach to knowledge sharing. Such a culture has to be embedded in the daily business operations in the working atmosphere. Every stakeholder of the business, such as managers, supervisors, workers or even the business owners must take the initiative to participate and create the knowledge sharing environment.

In process capability, Chan and Chao used four factors to represent the activities involved in knowledge operations. These four factors are acquisition, conversion, application, and protection. Acquisition process refers to how knowledge is acquired from various external and internal sources. The external sources may include business partners in the supply chain, government agencies, competitors, or organizations dealing with standardizations such as ISO or ANSI. Organizational policies, profit and benefit analysis are regarded as knowledge acquired from within the organization. Knowledge acquired from either external or internal sources is ineffective unless it is converted into useful and applicable forms to improve productivity and business operations. Therefore, conversion and application are important factors in process capability. The last factor is associated with knowledge protection. Rapid technology evolution and popularization has influenced most of SMEs to adopt the latest technology such as utilizing the Internet as the platform for hosting their knowledge asset. Through such an open platform, the organizations' knowledge is highly exposed by the public domains. The enterprises need to assure their organizational knowledge is kept safely and accessed only by authorized personnel. Protection of knowledge asset is an essential task in the organizational KM.

Traditionally, manufacturing sector is the major economic focus of Taiwanese SMEs. Through the technology evolution and changes of industry environment, Taiwanese government has been assisting and pushing SMEs to elevate their business activity towards service and knowledge oriented operations. Since July 2006, Small and Medium Enterprise Administration, Ministry of Economic Affairs in Taiwan established a four year "E-Oriented SMEs Plan". There are three main objectives through the E-Oriented SMEs Plan directly related to introduction and implementation of KM in SMEs. Firstly, developing knowledge sharing environment and platform to facilitate KM adoption, accumulate core business knowledge, and improve enterprise management efficiency. Until the end of 2007, there are 569 SMEs involved in business diagnosing and enterprise upgrading projects, which has created approximately US\$6 millions business opportunities in information systems and technology consulting. The plan has increased business growth by 15-25% in KM related products and services. Secondly, the plan helps SMEs to apply KM as part of business operations by increasing KM consultation capacity and service quality. There are 468 professional consultants involved in the consultation process. Thirdly, the plan concentrates on an expanding of the effectiveness for SMEs KM adoption. Through the E-Oriented SMEs Plan initiative, it is believed the SMEs have been providing assistance and opportunity by the government to exposure the benefit of KM adoption and exploitation.

3 A Unify Knowledge Management Model and Research Design

To conduct KM readiness study in Taiwanese SMEs, the authors have adopted Chan and Chao, and Gold et al's knowledge capability measurements as shown in Figure 1.

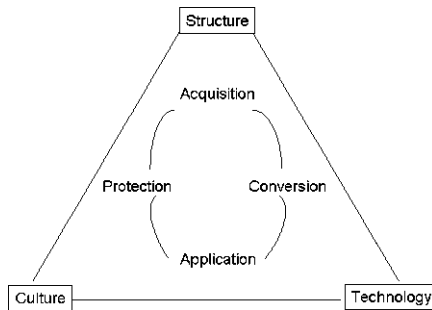


Fig. 1. A unify KM model [2]

The outer part of the figure represents infrastructure capability including technology, structure and culture. The inner part of the figure represents process capability including acquisition, conversion, application and protection.

With minor modifications, the survey instrument has been designed to collect data from SMEs in Taiwan in the six sections. Each of the first three sections consists of 7 items representing variables to the corresponding areas in technology capability, organizational structure, and organizational culture. The fourth section deals with KM process capability, and it is further divided into four subsections with a number of variables to measure four main process capabilities including acquisition, conversion, application, and protection. The fifth section highlights seven purposes for gathering the reasons why the SMEs to adopt and implement KM. The final section collects demographic data of the participating SMEs.

Our survey is drawn from 90 SMEs including 62 SMEs with KM initiative launched in the past few years. Respondents are from various industry sectors but are categorized under three main areas including high-tech (8%), manufacturing (36%), and knowledge services industry such as software development, innovation, and cultural (56%). Product and service type commodities are these SMEs main business focus, and they are more or less equally distributed (56% and 44%). With regarding to the company size, 26% have five or less of total employees; 30% have employees between 6 and 10; 15% have employees between 11 and 20; 19% have employees between 20 and 50; and 11% have total employee number between 50 and 200. Number of years the companies have been in business varies and spread across from less than one year (15%) to more than 10 years (15%). People responded to the survey are managers (2/3) and office administrators (1/3) of the organizations. These SMEs are mostly located in northern (63%) and middle (30%) parts of Taiwan.

Prior to conduct further statistical analysis, it is important to ensure the reliability of the valid respondents. A Reliability Analysis is used to test the reliability of the survey results. Reliability analysis enables the researchers to understand the components and properties of the measurements. It calculates a number of regularly used measurements reliability and extracts the information with regarding to the relationships between individual elements in the measurement. We adopt Alpha (Cronbach) model for the reliability analysis as this model allows testing the internal consistency, which is based on the average inter-item correlation. The results of reliability analysis are summarized in the following table (Table 1).

Table 1. Summary results of reliability analysis

Survey Section	Cronbach's Alpha Value
I. Technology Capability	0.598
II. Organizational Structure	0.764
III. Organizational Culture	0.887
IV. KM Process Capability	0.939

Based on the reliability analysis results, we are confident that all questionnaire measurements were conducted in a useful way. In technology capability, the Alpha value is between 0.5 and 0.6, which indicates the relationships of individual elements within this section are acceptable. In organizational structure, organizational culture, and KM process capability sections, the Alpha value of each section is between 0.7 and 0.9. This implies that a high acceptance and reliability of relationships between individual elements within each of these sections.

4 Data Analysis

The variables with the highest and lowest average respondent point (ARP) are calculated to explain the current KM practices and identify the issues to be resolved for a better KM adoption and implementation in SMEs. Each variable is measured by a typical five-level Likert scale as 1 – Strongly disagree, 2 – Disagree, 3 – Neither agree nor disagree, 4 – Agree, and 5 – Strongly agree. Thus the maximum ARP will be 5.

4.1 Infrastructure Capability

Technology. In Technology Capability, employees strongly agree that they are able to acquire important work related knowledge from the Internet and other electronic sources (ARP: 4.4). This indicates most of SMEs in Taiwan have incorporated the Internet access as part of their technology infrastructure. In contrast, the communication channels to facilitate knowledge sharing between the external entities (business partners and government agencies) and SMEs are not as frequent and convenient as getting knowledge from the Internet (ARP: 3.6).

Structure. From the Organizational Structure perspective, it is believed that many companies encourage knowledge sharing across business functions (eg. organizational departments and/or divisions) (ARP: 4.2). However, there are lesser SMEs (ARP: 3.7) agreed that their organizations have a common knowledge platform to enable employees to seek for work-related assistance. Hence, the SMEs need to draw attention in resolving the inconsistency between the company policy (encouragement of knowledge sharing) and the matching needs of enabling infrastructure (common knowledge sharing platform).

Culture. The primary concern of an effective knowledge sharing is “trust” between the participating entities. A clear understanding of organizational visions and objectives, and a high level of trust between the employees in relation to knowledge sharing are reported an average respondent point of 4.1. This indicates that many SMEs

have built-in such organizational culture in the daily operations and business strategies. An ARP of 3.4 for the companies providing sufficient support and training to the employees to increase the work efficiency is considered less than anticipated practice. It is imperative that the appropriate support and training programs should be in place when KM system is adopted and implemented in the SMEs.

4.2 Process Capability

Acquisition. Capturing knowledge from external and internal sources is a sophisticated process. It is even a challenging task to acquire quality knowledge at the right time and place. From the survey results, an ARP of 4.2 indicating that organization has the procedures to acquire new product/service and competitor related knowledge within the same industry sector. This is an encouraging result as it confirms that Taiwanese SMEs have such communication channels to capture competitive knowledge. There is also a high ARP (4.1) of organizations having current procedures to implement standardized guidelines for knowledge acquisition.

Conversion. In knowledge conversion phase, the results indicate a strong concurrence of organizations having current procedures to convert competitive intelligence to operational plan (ARP: 4.3), and transform knowledge from employees and business partners to its operations (ARP: 4.3). However, a less agreement of organizations having the procedures to promote the operational knowledge and transfer it to employees (ARP: 3.6). This matches the lack of training programs and support to the employees identified in "Culture" part of the infrastructure capability. It reconfirms that appropriate and essential training programs are necessary to ensure the employees (both new and existing) understanding and applying the pathways to receive organizational knowledge.

Application. Corporate knowledge becomes the most important intangible and invaluable asset only after it has been applied to the business operations and decision marking appropriately. It is rather a positive feedback that an ARP of 4.3 of organizations having the capability to utilize knowledge for solving new problems. But when it comes down to the implementation level, a lower ARP (3.7) is reported that the organization can rapidly supply the necessary knowledge to appropriate parties. It demonstrates that most of Taiwanese SMEs have the overall framework to deal with the application and exploitation of acquired knowledge. However, a lack of supporting mechanism, which allows required knowledge to be delivered in time for seamless business operation and decision making.

Protection. Security is always the major concern in any organization's management information systems. Protecting corporate knowledge requires clear but detailed policies to ensure the knowledge asset is in its safe state at all time (24/7). Through the survey results, Taiwanese SMEs are fully aware of the importance of the organizational knowledge protection and have procedures to manage unauthorized access (ARP: 4.3). Conversely, a lack of login and access policies in place to protect organizational knowledge is reported (ARP: 3.8). Once again, the detailed and systematic procedures with regarding to the organizational knowledge protection need to be spelt out at the operational level.

4.3 Purposes of KM Adoption

Looking at the KM adoption purposes, similar to Hong Kong SMEs (49.2%), managing knowledge resources is considered the main objective of pursuing KM in business operations (79%) in Taiwanese SMEs. There are more than 2/3 of the Taiwanese SMEs (68%) believe that KM implementation will enable the companies to gain competitive advantages whereas less than half (41.3%) of Hong Kong SMEs reported in this KM goal.

Over half of the SMEs suggest that KM operations will assist the organizations in improving business processes (57%) and reducing duplication of work (61%). One half of the SMEs (50%) recommend that the incorporated KM in business systems will increase the company profit and inspire innovation. These are quite different from Hong Kong SMEs, which have received least attention or are not embraced in their business agenda. An obvious explanation is that there are more manufacturing SMEs in Taiwan than in Hong Kong, and they foresee the greater benefits and innovation opportunity by involving in KM.

It is an interesting note that there are only 14% of Taiwanese SMEs considering KM implementation will provide controlling of information overload, which is consistent with Hong Kong SMEs. This leaves the researchers and practitioners an immediate future development agenda item of a practical mechanism to filter, validate, and maintain up-to-date knowledge, which will overcome the issues in redundancy and overload of knowledge.

5 Discussion

To compare KM adoption in SMEs in Taiwan and Hong Kong, the authors compiled the Taiwanese SMEs survey results and the study done by Chan and Chao [2], and summarized the KM adoption maturity level in Table 2. The table shows distinctive characteristics of KM in Taiwan and Hong Kong and highlights the KM adoption maturity level of each. The maturity level is divided into five categories: H (High 81-100%), HM (High-Medium 61-80%), M (Medium 41-60%), ML (Medium-Low 21-40%), and L (Low 0-20%). Each category is based on the analysis of the empirical data collected in both studies.

Table 2. KM adoption maturity level in Taiwan and Hong Kong

KM adoption maturity level	Taiwan					Hong Kong				
	H	HM	M	ML	L	H	HM	M	ML	L
Technology			X					X		
Structure	X						X			
Culture		X				X				
Acquisition		X					X			
Conversion		X					X			
Application		X					X			
Protection		X					X			

As from the table, the results in both Taiwan and Hong Kong studies are quite similar. The technology part of the infrastructure capability and all of the process capabilities are on the same maturity level for both economics. The differences are the organizational structure and culture, which Hong Kong has higher maturity level in culture aspect and less maturity level in structure. The comparison suggests that the KM adoption strategy and development in either Taiwan or Hong Kong can be easily transformed and applied in either of both economics.

Bohn [1] presented a method to measure and manage technology knowledge. Based on the unify KM model, Figure 2 shows a proposed KM growth structure based on infrastructure and process capabilities. If an organization does not do anything in process and infrastructure capabilities, then the organization is complete ignorance in KM. The awareness shows the organization aware the importance of knowledge, but not yet implemented. The process characterization means the organization understands very clearly about the organizational operation process. The infrastructure characterization shows the organization focus on infrastructure capability such as technology, structure and culture. The know-why knowledge represents the organization deployed KM. If both the infrastructure and process received a highest score represents the organization completes implemented KM. Based the survey results in Taiwan, the T symbol represents in Figure 2 shows that Taiwan SMEs in know-why stage of deployed KM. The average respondent point of infrastructure capability is 3.88 whereas the average respondent point of process capability is 4 in Taiwan SMEs in KM.

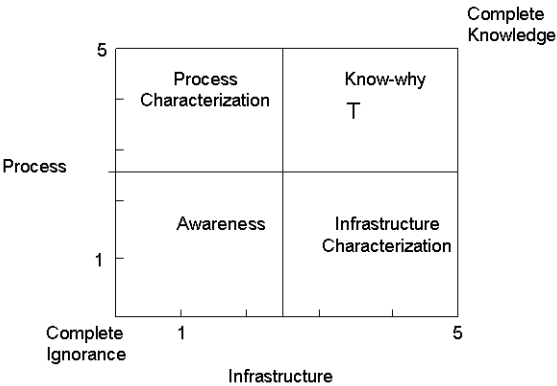


Fig. 2. KM growth structure

6 Conclusion

Adoption of KM in SMEs has become the emerging agenda in developing business strategies. In order to further utilize KM for seamless business operations and decision making, organizations must understand the benefit of implementing KM and its current status of KM readiness. This study examines the infrastructure and process capabilities of Taiwanese SMEs and compares the survey results with the similar study done in Hong Kong recently. As anticipated, the results are quite parallel in

both economics due to the SMEs in Taiwan and Hong Kong possess similar economic atmosphere and business background.

Through the empirical data analysis, the authors have drawn numerous important issues, which the SMEs need to address in the future development and implementation of KM. In infrastructure capability, organizations need to establish the communication channels to facilitate knowledge sharing with the external entities such as business partners and government agencies. It is suggested that the organizational policy in KM should be consistent with the activities in the operational level. Appropriate support and training programs for employees should also be in place when KM system is adopted and implemented in the SMEs. In process capability, once again, the appropriate and essential training programs are necessary to ensure the employees understanding and applying the pathways to receive organizational knowledge. Furthermore, the detailed and systematic procedures with regarding to the organizational knowledge protection need to be in place at the operational level.

Utilizing KM in controlling information overload in SMEs is still in its pre-mature stage. The authors recommend that a practical mechanism to deal with validation and accuracy of organizational knowledge is the next phase of KM evolution in SMEs.

References

1. Bohn, R.: Measuring and Managing Technological Knowledge. *Sloan Management Review* 36(1), 61–73 (1994)
2. Chan, I., Chao, C.: Knowledge management in small and medium-sized enterprises. *Communications of the ACM* 51(4), 83–88 (2008)
3. Denning, S.: Ten steps to get more business value from knowledge management. *Strategy & Leadership* 34(6), 11–16 (2006)
4. Gold, A.H., Malhotra, A., Segars, A.H.: Knowledge management: An organizational capabilities perspective. *Journal of Management Information Systems* 18(1), 185–214 (2001)
5. Griffith, T.L., Malhotra, A., Neal, M.A.: Virtualness and Knowledge in teams: Managing the love triangle of organisations, individuals and information technology. *MIS Quarterly* 27(2), 265–287 (2003)
6. Handzic, M.: Knowledge management in SMEs: Practical guidelines. *Asia-Pacific Tech. Monitor*, 21–34 (January-February 2004)
7. KM plan for small and medium enterprise. Small and Medium Enterprise Administration, Ministry of Economic Affairs (2006) (accessed October 28, 2008), <http://smekm.moeasmea.gov.tw/>
8. Nunes, B.M., Annansingh, F., Eaglestone, B., Wakefield, R.: Knowledge management issues in knowledge-intensive SMEs. *Journal of Documentation* 62(1), 101–119 (2006)
9. Tseng, S.: The effects of information technology on knowledge management systems. *Expert Systems with Applications* 35(2008), 150–160 (2007)

Supporting Strategic Decision Making in an Enterprise University Through Detecting Patterns of Academic Collaboration

Ekta Nankani, Simeon Simoff, Sara Denize, and Louise Young

University of Western Sydney

Abstract. Collaborative networks are a topic, broadly researched from several perspectives, including the social network analysis (SNA). The organisations take advantage from the results of SNA to determine collaborative channels, information fusion through such channels and key participants or groups in the network. This work is focused on multi-facet analysis of academic collaboration, as it has been identified as a key factor in success and growth in the global educational market. The data sets include integrated data about different aspects of academic collaboration, including co-authorship, co-participation, co-supervision and other related data. We explore the concept of interestingness and its application to the field of network mining. Composing an appropriate interpretable set of interestingness measures will benefit decision makers in organisations in taking specific actions depending on the patterns in these measures. In this study we focus on interesting measures such as unexpectedness for academic networks and a collaborative score.

Keywords: Collaboration, Academic Networks, Social Networks.

1 Introduction

Among various organisations universities occupy a special place as self-organising institutions. Over the last decades there have been changes in the universities in Australia, which have had an impact on the governance and decision making processes. These changes have led to the emergence of new kind of higher education institution, labeled by some authors as "The Enterprise University" [1]. This new kind of higher education institution resembles many elements of corporate governance. It has to take in account market factors, such as student fee incomes, soft budget allocations for special initiatives, including research funding, risk factors and others. This openness to external funding and competition is part of the changing global market in higher education. As *research* is a major component of academic activities and reputation of universities, the development of strategies for sustaining a distinct research profile is an essential task for the senior executive teams in these new type of institutions. As universities operate in underfunded environment and under resource constraints, understanding, utilising and strategically driving the structure of academic collaboration is a major

component of sustained research scholarship and enabling of the nexus between research and teaching, which makes each university a unique learning environment. Another for getting, monitoring and understanding current structure of academic collaboration in a university is that the most exciting and groundbreaking research in many cases emerges from the interaction and combination of several disciplines. Understanding the structure of existing and the potential of new predicted collaboration is also a key for fostering collaboration between industry and academia, which is seen as key driver of innovation both in education and technology. The development of business intelligence methods capable of reliable modeling of collaboration is essential for enabling senior executive teams in these new type of institutions for sound strategic planning.

1.1 Scientific Networks, as a Precursor to Collaborative Academic Networks

Collaborative networks are a topic, broadly researched from several perspectives, including social network analysis and mining. Organisations can take an advantage from the results of social network analysis when determining emerging collaborative networks, the information fusion through the links in such networks and the key participants and/or groups in these networks.

The study of scientific networks has been attracting an increasing attention. The main focus has been on the analysis of research works in terms of their authorship and citation. This led to three main types of analysis and respective network models: (i) co-authorship [9, 12], (ii) citation [15] and (iii) co-citation [4]. In co-authorship networks two researchers are considered connected if they have co-authored a research work (e.g. their names appear on the same document). The connection does not have a direction. In citation networks two researchers are considered connected if at least one of them has cited a research work (co)authored by the other author. The actual link is between the two documents, i.e. between the subject topics in the documents. The link then expands into link(s) between the authors of the documents. In general, the links in a citation network are directed. In co-citation networks two researchers are considered connected if one or more of their research works are cited simultaneously in the same document. As citation is a topic relation, again the links are between the topics, and the association between authors is assumed to be purely on the relatedness between the research topics rather than a collaborative activity.

These three types of analysis are currently underpinning the intelligence that is provided by many scientific portals and by visual analytics tools in the area. In terms of the deeper analysis of academic collaboration, the method discussed in this paper expands the above discussed co-authorship approach.

1.2 Information Sources and Visual Analysis of Academic Networks

As network models usually comprise of large amount of nodes (e.g. in a university that could be few tens of thousands of nodes), with the advent of visualisation algorithms and displays, it has become common to represent visually the network

Table 1. Information sources - type of data, source, typical location and analysis methods (for some sources)

Joint authorship	Data from Research Granting bodies, Academic Divisions and Personal websites
Joint supervision of HDR students	University division for graduate students
Interest in/Attendance at same conferences	Conference publications (longitudinal analysis)
Membership in a research center/division	Research centre/division websites
Congruence in semantics of research descriptions (common words)	Websites, papers, analysed by text analysis/mining algorithms
Third parties in common (weak links)	Conventional data sets, analysed by link analysis methods
Teaching collaboration	Data about subject delivery, time tables (time stamps)Curriculum development documentation
Citations of others work, mutual- and co-citation	Citation analysis (restricted to the identified networks)
Publication in similar journals (though not together)	Journal portals, Personal Websites
Who is perceived to be working in similar fields	Qualitative Interviews with Research coordinators in Schools and Units (3-5 per Faculty estimate)
Grant applications from different schemes (both successful and unsuccessful)	University Research Office; Funding bodies portals
Members of Common Professional/Research Associations, Conference Committees/Co-Chairs	CVs/ websites
Working paper (including perhaps publishing papers in the same series)	University Web Site
Email data: Reciprocity	Desensitised data from Email servers - only message headers
Common Academic, Governance and other Committee membership and working parties	University intranet data

models and then to apply some form of visual analysis. In order to provide reliable support to the decision making process, the proper analysis of network models requires the ability to extract patterns at different levels of granularity. This requirement translates into requirements for high volume data collection, processing, mining, modelling and communicating the models quickly to the decision makers.

As academic activities comprise from research, teaching and various other services, an accurate analysis of these activities requires linking (if not integrating) data coming from diverse sources. Table 1 shows the diversity of information required for the analysis of collaboration, the data sources and some of the methods for information extraction from those data sources. The list is limited to data, which collection is part of the operational processes and its analysis to a certain extent can be automated.

Without loss of generality in terms of presenting the approach and methodology, in this paper we have focused only on research collaboration limiting the data set mainly to data about: - publications, where publications are classified in the following categories: book, book chapters, journal and conference papers; - research projects, where participants are classified in the following categories:

internal academic staff, students and external participants; and - data about the supervision of higher degree research students.

Proposed approach analyses the network at different levels of granularity, varying from individual level through to networks between divisions. The different levels of networks indicate the author level, department level and school level.

The paper is organised as follows: Section 2 looks at the measures of interestingness and their relation to collaborative networks; Section 3 presents the approach and methodology on the example of academic collaboration within a University. Section 4 discusses the results of the analysis. Section 5 considers the limitations of the approach, future developments and concludes the paper.

2 Interestingness and Aspects of Measuring It

Interestingness of discovered patterns is a key concept with respect to the decision making process. The concept of interestingness in analytics encapsulates "conciseness, reliability, peculiarity, diversity, novelty, surprisingness, utility, and action ability" [2]. In the area of mining social networks, the concept of interestingness has been considered only recently, with more emphasis on the action ability side - the benefit in organisational context comes from the specific actions, based on the rankings of the outcomes, subject to the interesting measures. Measures of interestingness are the technique to decrease the generated patterns and increase relevance of the discovered patterns [10].

Mining data about collaboration networks yields numerous patterns and essential remains the detection of the "most interesting" patterns for the analyst. In order to do that the generated patterns are evaluated using some interestingness measures for ranking the discovered knowledge that signifies the user requirements [13]. Interestingness can be measured with unsupervised link discovery methods using the notion of rarity and abnormality. Rarity is indicative to interestingness as such patterns do not exist frequently and abnormality is a relative measure of abnormal connection between nodes [10].

The measures of interestingness are classified as objective and subjective measures [5]. The objective measures depend on patterns and underlying data - according to McGarry "The objective approach uses the statistical strength or characteristics of the patterns to assess their degree of interestingness" [7]. Subjective measures are defined by the class of analysts - "subjective techniques incorporate the users subjective knowledge into the assessment strategy" [7]. Major focus in this paper is on subjective measures, which are related to the system of beliefs. In order to determine interestingness a belief system is defined based on the academic domain knowledge [11]. Though the main goal of this research is inclined to gauge the interestingness in the network, we have also made an attempt to compute the knowledge volume and the richness of knowledge transferred. Researchers, involved in studying international scientific collaboration between countries and the collaborative networks associated with scientists employed different methods for measuring associations in collaborative

networks, including Salton's and Jaccard's measures and the Pearson's product moment correlation [6].

Few of interesting measures are the Similarity measures, which are used for making recommendations in the online information services and the social networking sites [5]; the Shocking rules, another subjective measure of interestingness, these rules are never expected and are very novel as they do not exist in the previously discovered knowledge; the KAFIR system that consider "a good measure of the interestingness of a finding is the estimated benefit that could be realized by taking specific action in response" [13]. Also the interestingness of a measure depends on the group of analysts analysing the data as some of the measures that are interesting to one analyst may not necessarily be of any interest to the others.

The subjective measures generate results that can be unexpected, actionable and novel. The patterns or rules are classified as unexpected if they are surprising to the analyst; they are actionable if the decision maker can act to their advantage, the degree of actionable depends on the application; and they are novel if they contribute to new knowledge [3]. According to Padmanabhan and Tuzhilin (1999) "a rule is considered to be unexpected if it intuitively, "shakes" the system of beliefs, including the changes to the degrees of these beliefs." As subjective measure depends on both the data and the user, it requires the domain specific knowledge. The user's domain knowledge and experience could add value to the knowledge discovery process [11].

2.1 Interestingness in Collaborative Networks

The measures of interestingness for co-authorship networks, as discussed by Newman [9], are related to the shortest paths between nodes. Interestingness in such cases is expressed through centrality measures, in particular, Betweenness, it indicates the influence that individuals have over information flow between others. Two other measures considered interesting are the clustering coefficient or transitivity that measure the probability that two nodes related to some nodes have collaborated; and "assortativity coefficient" or degree of correlation, this is the correlation coefficient for the number of collaborators a node has [9]. The other measures considered as measures of interest are indegree, outdegree of the nodes; joint degree distribution that provides many insights of the structure of the social network [8]. Worth mentioning are the similarity measures over network models. These measures can be bilateral and multilateral. Bilateral measures, such as Salton's and Jaccard measures concentrate on two nodes at a given time. These measures may have bias against some nodes and underestimate the links between them. While multilateral similarity measures take the whole network into account, such as Goodman quassi-independence model, which calculates the ratio between the observed and expected volume of links [6].

We have used measures of interestingness in the visual analysis of the networks in order to select projections and visual drill down/drill up operations. Rank/degree of collaboration can be calculated as a function of the percentages of activities between the academics involved. The study that we use to illustrate

the approach, analyses the interesting measures such as unexpectedness in academic networks and proposes a collaborative score, calculated between the actors, in order to identify key collaborating entities.

3 Extracting Information about Collaboration

This example is based on a university data to investigate the interesting patterns, measures and collaboration between actors at different levels of granularity.

3.1 Data Set

Table 2 shows the description of the data set, which includes integrated data about different aspects of academic collaboration, including co-authorship, co-participation in research project, co-supervision of research students and other related data. This is a time series data collected over a consecutive span of 5 years. The university has 9 schools and 23 research centres. All the collaborative ties are between staff, students and externals. The visual analytics techniques

Table 2. Description of the data set

<i>Description</i>	<i>Number of Records</i>
Data Records collected	24,556
Clean Data Records	15,177
Number of Distinct Nodes	2,131
Number of Ties	37,398

used to analyse the data, is illustrated with the application of the NetDraw graph visualisation system [16]. The NetDraw support the visualization changes in the networks over a time span, generates the centrality measures and facilitate understanding of the social networks.

3.2 Methodology

Figure 1 shows the process flow of the data preparation and the analysis in the example. It is part of the adaptation of the industry standard CRISP-DM data mining methodology [14]. The main steps include:

- Academic business understanding;
- Collaborative data integration and analysis for developing data understanding;
- Social network mining tasks - creating the social network structures, estimating various network statistics and measures of interestingness, interactive visualisation of the network structures, visual drill-down/drill-up operations on the network structures;
- Network model analysis that includes calculation of collaboration score and evaluation of the structures, multiple network analysis.

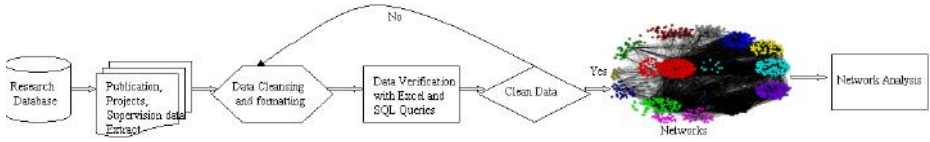


Fig. 1. Process flow of the data preparation and analysis

3.3 Analysis

We use the following notation and terminology.

1. Actor: An actor in the analysis depends on the level of the analysis - it can represent a staff member, student, an external member, or an academic department, an external organisation.
2. Tie: A collaborative activity described in the data set, such as co-authorship, co-participation on a project, co-supervision of higher degree research students.
3. Collaborative Activity (*CollabAct*): Co-authorship, Co-supervision and Co-participation.
4. Contribution Score (*ContScr*): A measure of the individual actor contribution to a collaborative activity.
5. Total Contribution Strength (*TotContScr*): The total contribution made by an actor.
6. Collaboration Score (*CollabScr*): Collaboration value scored by an actor for a given dataset.

Collaboration Score. In this study we look at the level of participation in a collaboration as a function of the participation in the individual activities. For each actor we calculate a *collaboration score* *CollabScr* which signifies the strength of the contribution to the collaboration. The contribution score $ContScr_A$ for a Collaborative Activity *CollabAct* of an actor *A* is defined as

$$ContScr_A = \frac{1}{\text{number of participants in the } CollabAct}$$

The sum for all contributions ($ContScr_A$) for all collaborative activities *CollabAct* in which actor *A* participated is defined as

$$TotContScr_A = \sum ContScr_A$$

Then the Collaboration Score for actor *A* is defined as

$$CollabScr_A = TotContScr_A * \frac{n}{N}$$

where $n = \text{Count of Actor } A \text{ } CollabAct$ and

$N = \text{Total count of collaborative activities in the dataset.}$

Table 3 shows the top 4 actors with high *CollabScr* values.

In the next section we have discussed the egonets for all these top four actors with highest Collaboration Score.

Table 3. The top four actors in terms of collaboration score

Actor Name	CollabScore
A	1.17922
B	0.63753
C	0.65718
D	0.54995

Circle		External
Up Triangle		School of IT
Box		School of Business
Down Triangle		School of Design
Circle-in-Box		School of Education
Diamond		School of Engineering
Plus		School of Social Science
Circle		School of Law
Rounded Square		School of Nursing
Thing		School of Science
Circle		Research Center
Box		Student

Fig. 2. Legend

Network Structure Analysis. The Figure 2 illustrates the colour coding scheme used for different school and the different shapes used to represent different schools in the Figure 7.

In the Figure 3a, the network depicts an overall network including all the ties among individuals and the individuals are grouped under the schools; the Figure 3b is the network where all the individuals are collapsed into their respective departments within the school; the Figure 4 is the network where all the individuals are collapsed into their respective schools. The network show the peculiarities: The dense Students and External nodes in the Figure 3a reflect there intense participation in the collaborative activities; Figure 3b shows School of IT, School of Business and School of Engineering have strong links of collaborations within the schools; Figure 4 shows School of Engineering, School of Business and School

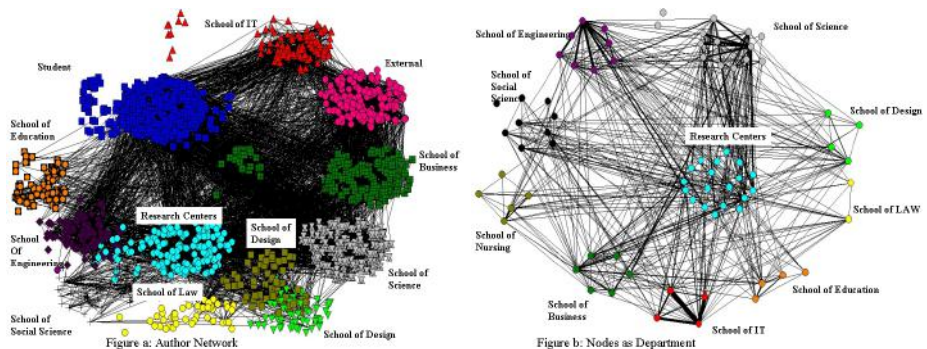


Fig. 3. Initial Networks

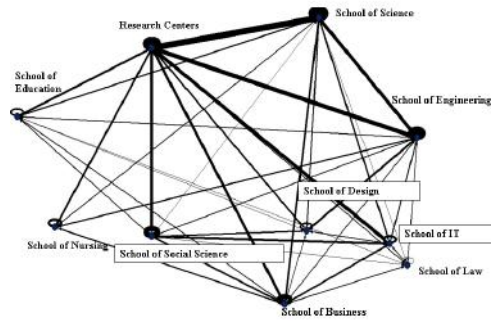


Fig. 4. Network based on School

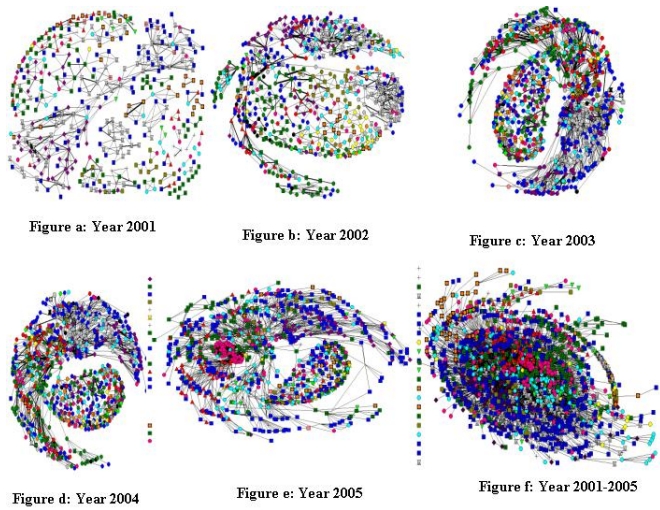


Fig. 5. Year Wise networks

of Science have collaborated with all the other schools in the university; Figure 4 also shows School of Science have collaborated the most with Research centers; Figure 4 display all the schools are connected with Research centers some of them are densely connected other are sparsely connected.

The Figure 5 , display networks based on slices of collected data ranging from the year 2001 data (Figure 5a), 2002 data (Figure 5b), 2003 data (Figure 5c), 2004 data (Figure 5d) and year 2005 data (Figure 5e). The last network is based on the time span of five years from 2001-2005 (Figure 5f). By visualising the networks in Figure 5, suggest that the *collaborative activities and the participants* in the university for each school gradually increased between the period 2001-2005.

The Figure 6, shows the egonet for top 4 actors with highest collaboration scores. It can be identified from the egonet networks that Actor A and Actor C have strong links with externals.

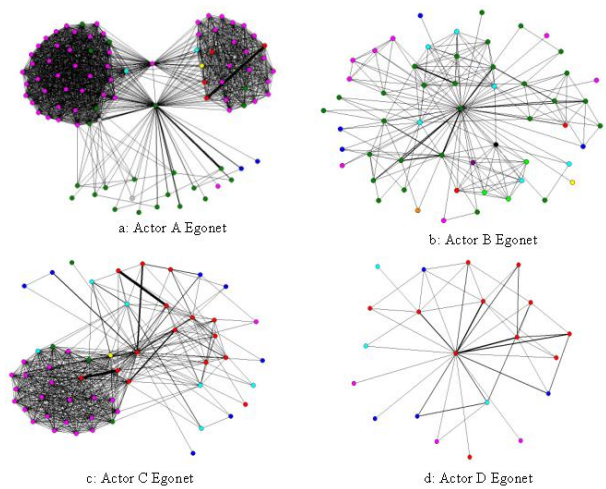


Fig. 6. Egonet

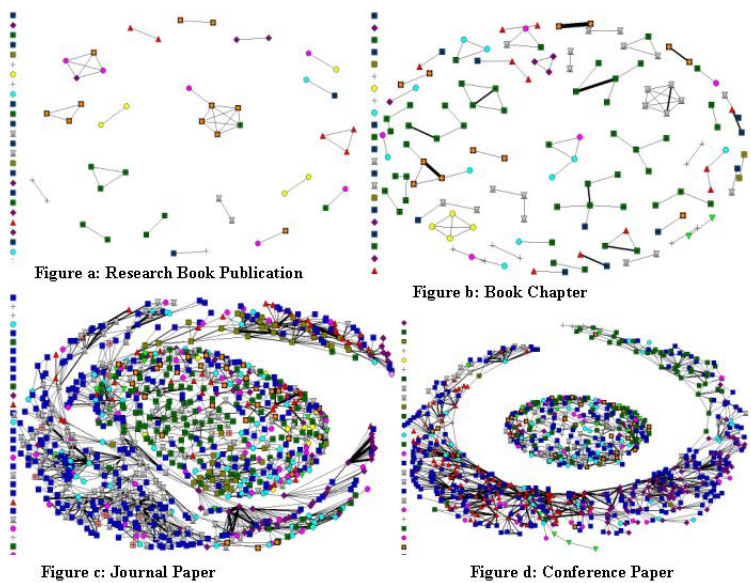


Fig. 7. Networks based on Publication Category

The Figure 7, display the network based on DEST Publication category, we have selected 4 categories. The Figure 7a is category A1 publications. These are Research Book publication network, the network structure suggest that most of work for this category of publication is done within the school. The Figure 7b is category B1 publications. These are Book chapter publication network,

the network structure suggest school of business has participated actively for Book chapter publications. The Figure 7c is category C1 publications. These are Journal article publication network, the network structure suggest that students, school of business and school of science has the most participation. The Figure 7d is category E1 publications. These are Conference article publication network, the network structure suggest that students, school of business and school of IT has largely contributed.

4 Conclusions and Future Work

This paper display the collaborative networks present in the university using time series data for collaborative events. As in the global economic picture with increased competitiveness under resource constraints, universities success and growth depends largely on collaborations within them and with outside partners. Our study identifies the key players, schools and departments in the research field. It also shows the increase in collaborative activities over a period of time. We have proposed a metric to measure the collaboration statistically with the help of collaboration score *CollabScr*.

The next stage of detecting patterns of collaborations is to study the richness of the knowledge transfered during collaborative activity. This can be achieved by collecting the explicit data as well as the tacit data with the help of interviews with actors, and survey questionnaires. The project co-participation can be analysed by the people generated publications out of the projects. Further networks can be learnt by slicing data for chief investigators; principal supervisors; individuals with high *collaboration scores* and performing h index [17] analysis for them.

References

1. Marginson, S., Considine, M.: The Enterprise University Power, Governance and Reinvention in Australia. Cambridge University Press, Cambridge (2000)
2. Geng, L., Hamilton, H.J.: Choosing the Right Lens: Finding What is Interesting in Data Mining. In: Guillet, F., Hamilton, H.J. (eds.) *Quality Measures in Data Mining*, pp. 3–24. Springer, Heidelberg (2007)
3. Cooley, R., Tan, P.-N., et al.: Discovery of Interesting Usage Patterns from Web Data. In: Masand, B., Spiliopoulou, M. (eds.) *WebKDD 1999*. LNCS, vol. 1836, pp. 163–182. Springer, Heidelberg (2000)
4. Small, H.: Co-citation in the scientific literature: A new measure of the relationship between two documents. *Journal of the American Society for Information Science* 24(4), 265–269 (1973)
5. Hilderman, R.J., Hamilton, H.J.: Knowledge discovery and interestingness measures: A survey (1999)
6. Luukkonen, T.R., Tijssen, J.W., et al.: The measurement of international scientific collaboration. *SpringerLink* 28(1), 15–36 (1993)
7. McGarry, K.: A survey of interestingness measures for knowledge discovery. *The Knowledge Engineering Review* 00, 1–24 (2005)

8. Mislove, A., Marcon, M., et al.: Measurement and analysis of online social networks. In: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, San Diego, California, USA. ACM Press, New York (2007)
9. Newman, M.E.J.: Coauthorship networks and patterns of scientific collaboration. PNAS 101(1), 5200–5205 (2004)
10. Shou-de, L., Hans, C.: Using unsupervised link discovery methods to find interesting facts and connections in a bibliography dataset. SIGKDD Explor. Newsl. 5(2), 173–178 (2003)
11. Silberschatz, A., Tuzhilin, A.: On Subjective Measures of Interestingness in Knowledge Discovery. In: First International Conference on Knowledge Discovery and Data Mining. Palais Des Congress, Montreal, Canada. AAAI Press, Menlo Park (1995)
12. Glänzel, W., Schubert, A.: Analysing Scientific Networks through Co-Authorship. In: Moed, H.F., Glänzel, W., Schmoch, U. (eds.) Handbook of Quantitative Science and Technology Research, pp. 257–276. Springer, Heidelberg (2004)
13. Yafi, E., Alam, M.A., et al.: Development of Subjective Measures of Interestingness: From Unexpectedness to Shocking. PWASET 26, 368–370 (2007)
14. Chapman, P., Randy Kerber, J.C.S., Thomas Khabaza, T.R.D., et al.: CRISP-DM 1.0, Step-by-step data mining guide (2000), <http://www.crisp-dm.org/CRISPWP-0800.pdf>
15. MacRoberts, M.H., MacRoberts, B.R.: Problems of citation analysis. Scientometrics 36(3), 435–444 (1996)
16. NetDraw Network Visualisation, <http://www.analytictech.com/netdraw/netdraw.htm>
17. Hirsch, J.E.: An index to quantify an individual's scientific research output (2005), arXiv:physics/0508025v5 [physics.soc-ph]

Dynamic User Modeling for Personalized Advertisement Delivery on Mobile Devices

Luca Paolino¹, Monica Sebillo¹, Genoveffa Tortora¹,
Alessandro M. Martellone², David Tacconi², and Giuliana Vitiello¹

¹ DMI, Università di Salerno, Italy

{lpaolino,msebillo,tortora,gvitiello}@unisa.it

<http://www.dmi.unisa.it>

² Futur3 srl, via A. Abondi 37, 38100, Trento, Italy

{a.martellone,d.tacconi}@futur3.it

<http://www.futur3.it>

Abstract. In this paper we present an approach for the presentation of personalized advertisements on mobile devices, which is based on a user model that takes into account user's interests over time. The approach has been adopted within the LUNA wireless network project, which is targeted at realizing a business model such that the services provided in the area of Trento, in the North of Italy, are accessible and usable by everybody, at a very low cost. LUNA client-side mobile application relies on a dynamic user modeling technique. We describe the personalization component of the advertising management system employed and we explain how a user model is dynamically created, saved and updated on the basis of the latest interaction history and of the delivered contents.

1 Introduction

With the advent of broadband connections, Internet service providers have begun to charge users with fees in order to cover the higher infrastructural costs due to the management/rental of broadband networks. However, the user is often left the chance for a free Internet connection, provided that he/she accepts to be invaded by banner ads, which arbitrarily, in terms of time and position, appear on the screen. In order to make effective the latter approach, Internet service providers have been studying the most appropriate advertisement supply policy, in order to satisfy companies who wish to advertise their products without annoying the user while he/she is interacting with the service. Personalized delivery of advertisement is recognized to be a promising approach to capture users' interests in certain domains. However, further attention still deserves the issue of finding an adequate thread-off between the requirements elicited from advertising companies and the degree to which a user may accept to be bothered while performing some task. This problem is even more complex when advertisements are presented on mobile devices, where the reduced screen size considerably limits user's capability to continue his/her task, so increasing the frustration caused by the presence of the *ad*.

In this paper we present an approach for the presentation of personalized advertisements on mobile devices, which is based on a user model that takes into account user's interests over time. The approach has been adopted within the LUNA wireless network project, which is targeted at realizing a business model such that the services provided in the area of Trento, in the North of Italy, are accessible and usable by everybody, at a very low cost. LUNA client-side mobile application relies on a dynamic user modeling technique. We describe the personalization component of the advertising management system employed and we explain how a user model is dynamically created, saved and updated on the basis of a suitable combination of long term and short term user's interests.

The remainder of the paper is organized as follows. Section 2 contains a description of the LUNA project. In Section 3 we describe how a dynamic user model can be derived from a suitable combination of long term and short term user interests. Section 4 concludes the paper give a brief perspective on future work.

2 The LUNA Project

The LUNA project is a research project funded by the Province of Trento, through which the Italian company Futur3 aims at realizing a wireless network infrastructure able to provide access to everybody, everywhere and at any time. In order to make this possible in brief time at affordable costs, LUNA relies on emerging networking technologies, like WiFi and Wireless Mesh Networks, Hiperlan and lately Wi-Max. In particular, WiFi coverage will be provided for the cities of Trento, the cultural, administrative, research centre, of Rovereto, another important cultural and administrative location, and Riva del Garda, a highly touristic area. The deployed network is a Metropolitan Mesh Access Network named RAMM. Futur3 will provide services for its mobile customers, so that they can enjoy the WiFi access, with solutions bounded to the territory. This innovative network will offer also the opportunity to local actors to experience new technologies, increase interest for the area, and test innovative solutions for the citizen.

The services offered through LUNA network will be provided to users by the *Service Management* module that interacts with the *Profile Management*, the *Content Management* and the *Ads Management* modules, as shown in Fig. 1. Thus, the three main pillars of the service architecture are:

- 1) User Profile, i.e. user interests and user context, including its position within the network (localization).
- 2) Contents, i.e. all the information the users are requesting and getting back from the network.
- 3) Advertisements, i.e. all the commercial information that can be delivered through the network.

The way all the management components interact with each other is shown in Fig. 2.

The goal to keep users' connection to LUNA network and service fruition at a very low price is reached thanks to the "LUNA deskbar", a client-side service that users must maintain active in order to utilize the LUNA network. In fact, the main goal of the LUNA deskbar is to offer the user some additional service while pushing some advertising information on the her device, so that the connection can be kept at low price.

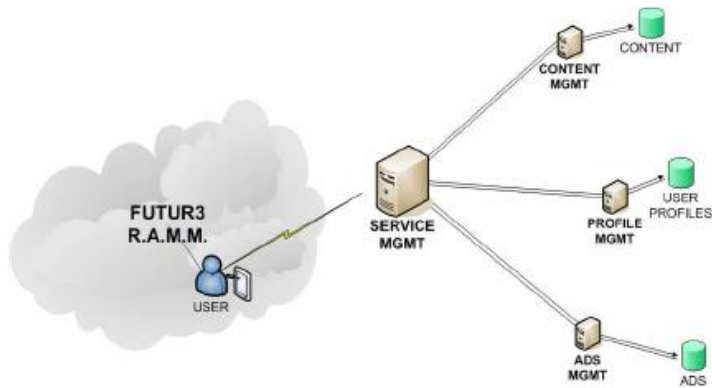


Fig. 1. Service Management Architecture

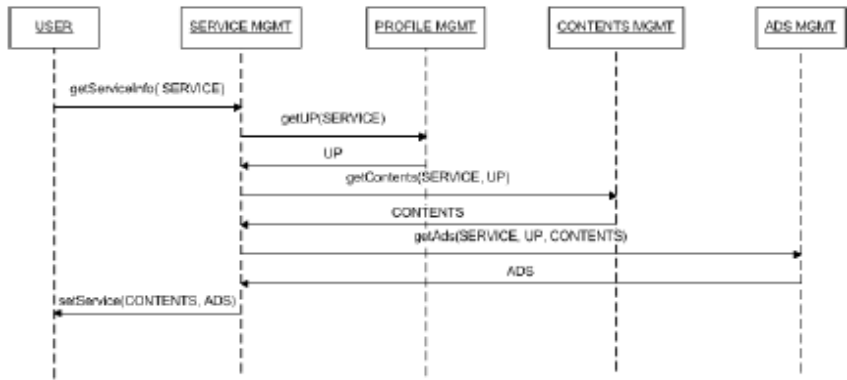


Fig. 2. Service activity diagram

3 A Dynamic User Model for Personalized Ads Delivery

The advertisements available for the publication on the mobile device through the LUNA system are chosen or discarded by associating them with a rank value which takes into account both long term and short term user interests and periodically verifies if the dynamically updated ranks are above a specific “pleasure” threshold or go below it.

In this section we describe the dynamic user model adopted in LUNA Ads Management module, and explain how long term interests and short terms interests are computed and then combined to derive a user-oriented rank.

3.1 Long Term User Interest Frameworks

Long terms user interests are modeled with respect to two reference frameworks: the first one based on an item domain classification and the second one based on advertisements contents mapped onto user’s general interests.

As for the first framework, when building a user profile, the system stores, for any item domain, the score the user assigns to any of the derived subcategories. For our purpose, we considered the classification of 35 product categories adopted by the *ebay*© platform. Thus, as an example, the domain *car* has the item subcategories sportive, limousine, city car, etc. When a user initially registers to LUNA services, he/she is asked to assign a score $S_{CAT}(u)$ to each of the general categories *CAT*, and a score $S_{Ci}(u_j)$, to any of its subcategories.

For any item domain, information concerning scores assigned by user to specific item subcategories are stored as a matrix, where rows correspond to the subcategories and columns corresponds to users (see Fig. 3).

CAT	u_1	...	u_m
C_1	val_{11}	...	val_{1m}
...	...	...	...
C_n	val_{n1}	...	val_{nm}

Fig. 3. User specified scores to CAT item subcategories

As each advertisement has a pre-assigned sub-category, selection with respect to this reference framework is immediate. Each advertisement a is assigned the score associated with the corresponding specific category in the corresponding user profile. Thus, the relevance to a user profile u of advertisement a , classified as belonging to subcategory C_i , is directly the score assigned to C_i by user u . Namely:

$$Rel_u(a) = S_{C_i}(u) \quad (1)$$

A similarity value $sim(a, CAT)$ is then computed between advertisement a and the general category *CAT* to which C_i belongs, exploiting the cosine similarity formula for the vector space model [1].

Consequently, the relevance between an advertisement a and all the general categories of a user model u is computed using the next formula:

$$REL_u(a) = \frac{\sum_{i=1}^{35} sim(a, CAT_i) S_{CAT}(u)}{\sum_{i=1}^{35} S_{CAT}(u)} \quad (2)$$

The second reference framework for long term interests, further specializes user's profile, based on a set of user-specified keywords, which are weighted based on their relevance to the user. These keywords are stored, for each user, as a term weight vector (k_u).

k_u	k_1	...	k_t
U	k_{1u}		k_{tu}

Again, the relevance between the advertisement and the keywords of a user model, we use the cosine for similarity within the vector space model:

$$rel_{K_u}(a) = sim(a, k_u) \quad (3)$$

Thus, the long term interest $LT(u,a)$ of a user u in a given advertisement a can be computed by combining formulas (1) to (3) as follows:

$$LT(u,a) = \frac{w_1 rel_u(a) + w_2 REL_u(a) + w_3 rel_{Ku}(a)}{w_1 + w_2 + w_3},$$

where w_1, w_2, w_3 are weights representing the importance assigned to the three relevance measures, referred to the specific domain category, to the general category and to user's keywords, respectively.

The value calculated on the basis of the long term user interests is successively updated combining it with a short term rank which is based on the feedback the user provides during the navigation.

3.2 Short Term User Interest Framework

Short term interests are represented by means of feedback terms. The terms are obtained from user provided feedback over the web documents he receives during browsing. That is to say, the user provides positive or negative feedback (f) over the document he receives, and a set of representative terms is extracted from them. This information is processed and the resulting value is a term weight vector (t). By fixing a p value, representing the number of the last web documents which we should consider for the short term value, we define:

$$r_{ad} = sim(a, t_d),$$

as the similarity degree between an advertisement a and the web document d , and:

$$r'_{au} = \frac{\sum_{i=1}^p f_i t_i}{p}$$

The r'_{au} value is taken to represent the current short term interests of that user. Short terms interests tend to correspond to temporary information needs whose interest to the user wanes after the connection session.

Another value which has been considered crucial to determine the short term interest of Luna user model is the location where the user is at the time the ad should be delivered. As a matter of fact, it seems reasonable that advertisements related to easily achievable places may be more interesting with respect to those far away. For this reason, we decide to involve the proximity value (pv) calculated as the inverse of the distance between the current position and the closest location where the advertised product could be purchased (see Fig. 4).

Thus, the short term interest $ST(u,a)$ of a user u in a given advertisement a can be computed as the sum of r'_{au} and the corresponding proximity value $pv(a,u)$.

By combining long term and short term interests, the total relevance between an advertisement a and a user model u is computed with the following formula:

$$Int(u,a) = \frac{w_1 rel_u(a) + w_2 REL_u(a) + w_3 rel_{Ku}(a) + w_4 ST(u,a)}{w_1 + w_2 + w_3 + w_4},$$

where w_4 is the weight representing the importance given to short term interest in the given user model.

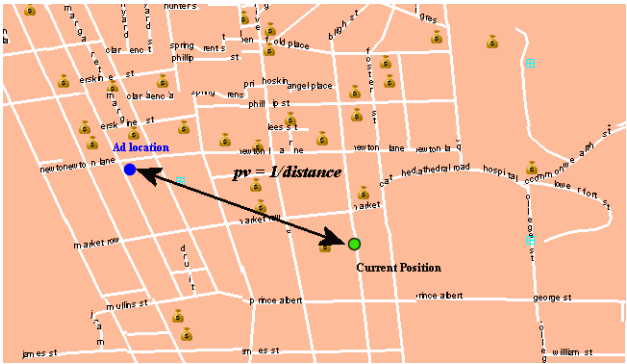


Fig. 4. User's proximity value wrt an *ad*

4 Conclusion and Future Work

In this paper we have faced the issue of establishing an adequate thread-off between the requirements elicited from advertising companies and the degree to which a user may accept to be bothered while performing some tasks. In the specific case of a wireless service network explicitly conceived to reduce access costs by means of adverts delivery, we have proposed a dynamic user modeling technique that may effectively support personalized ad delivery, within a ubiquitous interaction style. In the near future, we plan to perform experimental studies meant to validate/ properly calibrate the proposed approach. We will exploit specific usability testing equipment that has recently enriched the HCI laboratory of the Department of Mathematics and Computer Science, where some of the authors work. The equipment includes different samples of ubiquitous devices (including an Apple i-phone), and an advanced eye tracking system, that will allow us to derive reliable data from the experiments we will perform involvin a meaningful sample of LUNA users population.

Reference

1. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley, Reading (1999)

A Graphical Query Language for Querying Petri Nets

Lan Xiao, Li Zheng, Jian Xiao, and Yi Huang

Department of Industrial Engineering, Tsinghua University, Beijing, China
xiao102@mails.tsinghua.edu.cn, lzheng@tsinghua.edu.cn,
xiaojian99@mails.tsinghua.edu.cn,
huangyi00@mails.tsinghua.edu.cn

Abstract. As the number of business process models increases, providing business analysts and IT experts with a query language for querying business process models is of great practical value. This paper uses Petri net as business process modeling language and develops Petri Net Query Language (PNQL), a graphical query language for Petri nets. The syntax and semantics of PNQL are formally studied. PNQL allows users to get not only the perfectly matched Petri nets but also the Petri nets with high similarity. The complexity of PNQL is studied.

Keywords: Business Process Modeling, Petri Net, Query Language.

1 Introduction

Business Process Modeling (BPM) in systems engineering and software engineering is the activity of representing both the current ("as is") and future ("to be") processes of an enterprise. Business process models are created with an objective to capture business requirements, enable a better understanding of business processes, facilitate communication between business analysts and IT experts, identify process improvement options and serve as a basis for derivation of executable business processes.

As the number of business process models increases, providing business analysts and IT experts with a query language for querying business process models is of great practical value. This paper uses Petri net as BPM language and develops Petri Net Query Language (PNQL), a graphical query language for querying Petri nets. PNQL is based on a similar graph-based view of Petri nets, which is called model pattern. PNQL is provided with two types of semantics, which allow users to get not only the perfectly matched Petri nets but also the Petri nets with high similarity.

The rest of the paper is organized as follows. In section 2, we review related works. In section 3, we introduce Petri net as our BPM language. We present the syntax and semantics of PNQL in section 4 and study its complexity in section 5. These sections are followed by conclusions.

2 Related Works

Business process model query has been studied by several researchers. Most of them provided business process model with semantic annotations, which are the basis of the

model query mechanism. Cao and Zhao [3] proposed a semantic model for service workflow and its query mechanism. The semantic model for service workflow is based on the concept of goal. Wang et al. [20] put forward an extendable and configurable semantic framework to describe and query workflow with semantic template based on the goal ontology. Bowers and Ludascher [1] presented a calculus and two inference algorithms to automatically propagate semantic annotations through workflow actors described by relational queries. Markovic and Pereira [13] proposed an approach to business process modeling through reuse of existing business process artifacts - process fragments. In addition, they provided a rich formalism for business process description based on π -calculus and ontologies as a basis of the approach. Goud et al. [8] described a web application for managing collections of Petri nets. In the method, Petri nets can be assigned different kinds of properties: structural properties, e.g. the number of places; behavioral properties, such as boundedness or liveness; and metadata. And the properties can further support Petri net query. However, little work has investigated the query mechanism based on the structure of business process models. Oberweis and Sanger [14] put forward a graphical query language for large Petri net simulation runs. In the method, each query is made up of connected predicates and checkpoints, which represent sequence relationships between local markings. Nevertheless, the method is illustrated by two simple examples and no formal analysis is given.

In our framework, the similarity between business process models and model patterns provides important information for business process model query. Business process model comparison has been discussed in some previous works. Wang [19] proposed a method based on the process strings to compare two IDEF0 models. Whereas, this measure considers only the processing mode. The analysis of interrelationship between activities is omitted. Juan [9] developed an algorithm to search the alternative process paths in a flowchart and measure the similarity degree of activities contained in compared process paths. Juan and Ou-Yang [10] applied string coding and comparison to analyze the process logic differences between business process models. Nevertheless, these two methods of comparison are based on selected process path pairs, not on the whole model. Our framework provides a more comprehensive mechanism to compare business process models and model patterns.

Our framework is inspired by previous works on query languages for XML and ontology. Perez et al. [16] addressed systematically the formal study of SPARQL and provided a compositional semantics. Keramopoulos et al. [11] presented a graphical query language for object-oriented data base systems and provided its query mechanism. Comai et al. [4] described a graphical query language for XML data and gave an operational semantics based on the notion of graph matching. Braga et al. [2] presented a visual query language for expressing a large subset of XQuery in a visual form.

3 Petri Net

Recently, Petri net has been considered as a main tool for BPM [7,23,24]. There are a number of reasons for using a Petri net-based approach in modeling and analyzing business processes [15,22]. Petri net allows a graphical representation to ease the understanding of the modeled system and at the same time, they can be used in formal

analysis, verification and validation of the model [23]. Dussart et al. [11] compared several workflow modeling methods such as Petri net, WfMC, UML, ANSI, and EPC on criteria such as formal basis, executability, ease of visualization, etc. Their study showed that Petri net satisfied most of the criteria, and were therefore desirable. We also choose Petri net to model business processes. However, the query concept proposed in this paper is independent of modeling techniques, and not limited to Petri net.

A business process model may be defined at a structure level that depicts the flow relations between activities or at a behavior level that give its dynamic semantics. This paper only focuses on the structure level of business process models, which can also serve as a basis to further give the behavioral definition. As we are only concerned about on the structure level of business process models, we neglect the marking concept and firing mechanism of Petri net. The definition of Petri net is given as follows.

Definition 1 (*Petri net*). A Petri net is a triple (P, T, F) :

- P is a finite set of places
- T is a finite set of transitions ($P \cap T = \emptyset$)
- $F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs

For an arc $f = (v_1, v_2) \in F$, v_1 is its initial vertex, and v_2 is its terminal vertex. The place set, transition set, and arc set of a Petri net PN are denoted as $PN.P$, $PN.T$ and $PN.F$ respectively.

In general, transitions are used to represent activities. Places can represent the states to trigger activities or resulted from activities. For the sake of brevity, in the remainder of the paper, both transitions and places can also be called vertices. Activities and states represented by transitions and places can also be called events. Arcs between vertices represent the logical relations between events.

Petri net can be represented graphically. As in Figure 1, rectangles represent transitions, eclipses represent places, and directed lines represent arcs.

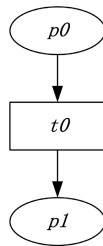


Fig. 1. Legend of Petri net

Routing structures, such as sequential structure, conditional structure, parallel structure and iterative structure, play an important role in BPM. Although Petri net does not have any notion of these structures, we can use Petri net to model these commonly used routing structures, as shown in Figure 2.

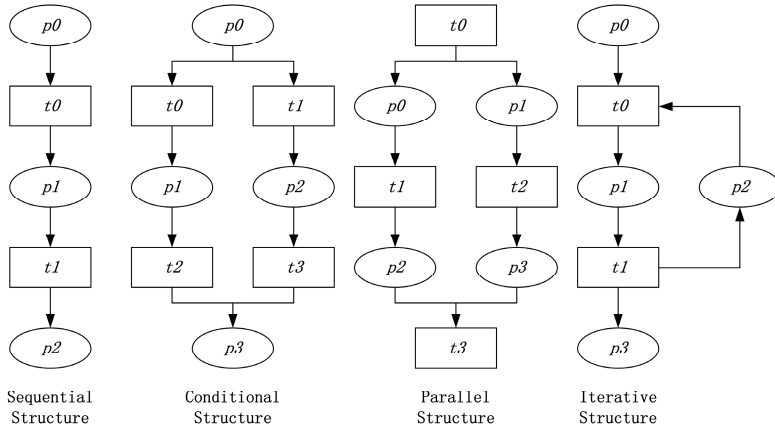


Fig. 2. Routing structures as Petri net representations

4 Petri Net Query Language (PNQL)

4.1 Preliminaries

To define the syntax and semantics of PNQL, we need to introduce some terminology.

Definition 2 (Adjacency Matrix). A matrix is the adjacency matrix (AM) of the Petri net $PN = (P, T, F)$ if and only if

- the $(i, j)^{th}$ entry of the matrix is equal to 1 if $(v_i, v_j) \in F$.
- the $(i, j)^{th}$ entry of the matrix is equal to 0 if $(v_i, v_j) \notin F$.

When the cardinality of $P \cup T$ is equal to n , i.e., $|P \cup T| = n$, AM is an $n \times n$ zero-one matrix. The objective of AM is to determine whether two vertices have an arc between them.

Definition 3 (Path). A sequence of arcs $(v_0, v_1), (v_1, v_2), \dots, (v_{n-1}, v_n)$ is a path in the Petri net $= (P, T, F)$ if and only if

- The terminal vertex of one arc is the same as the initial vertex of the next arc in the path.
- The arcs of the path belong to the arc set of the Petri net.

The path $(v_0, v_1), (v_1, v_2), \dots, (v_{n-1}, v_n)$ is a path from vertex v_0 to vertex v_n . The path can be denoted by $(v_0, v_1, \dots, v_{n-1}, v_n)$. The length of the path is n . v_0 and v_n are called the initial and terminal vertex of the path respectively.

Paths can provide important information about the relations between vertices. If there is a path from one vertex to another vertex, that is to say, under certain condition one event may occur before another.

Path matrix (PM) can be used to find out the number of paths from one vertex to another. The definition of PM is given as follows.

Definition 4 (*Path Matrix*). The path matrix (PM) of the Petri net $PN = (P, T, F)$ is equal to

$$\sum_{i=1}^{n-1} AM^i, \text{ where } n = |P \cup T|.$$

According to graph theory, if we want to traverse a Petri net with n vertices, the number of steps will be no more than n [17]. And the number of different paths from vertex v_i to vertex v_j of length no more than n is equal to the $(i, j)^{\text{th}}$ entry of PM .

Based on PM , path set (PS) can be derived. The definition of PS is given as follows.

Definition 5 (*Path Set*). (v_i, v_j) belongs to the path set (PS) of the Petri net $PN = (P, T, F)$ if and only if the $(i, j)^{\text{th}}$ entry of PM is greater than 0.

(v_i, v_j) belongs to the PS means that there are paths from vertex v_i to another vertex v_j . The PS of a Petri net PN is denoted as $PN.PS$.

4.2 Syntax and Semantics of PNQL

PNQL is based on a similar graph-based view of Petri nets, which is called model pattern. The definition of model pattern is defined as follows.

Definition 6 (*Model Pattern*). A model pattern is a 4-tuple $MP = (P, T, DF, IF)$

- P is a finite set of places
- T is a finite set of transitions ($P \cap T = \emptyset$)
- $DF \subseteq (P \times T) \cup (T \times P)$ is a set of direct arcs
- $IF \subseteq (P \cup T) \times (P \cup T)$ is a set of indirect arcs

Model pattern can be represented graphically. As in Figure 3, rectangles represent transitions, eclipses represent places, real directed lines represent direct arcs, and dotted directed lines represent indirect arcs.

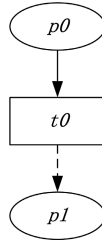


Fig. 3. Legend of model pattern

The semantics of direct arc is that there is an arc connecting the two vertices. The semantics of indirect arc is that there are paths between the two vertices.

The semantics of model pattern is defined as an evaluation over a set of Petri nets. Evaluation is a function which takes a model pattern as input and returns a set of Petri nets. Model pattern can be given two types of semantics. The first type of semantics requires that every Petri net in the evaluation result fully contain the information embedded in the model pattern. The second type of semantics allows Petri nets in the

evaluation result only partially contain the information embedded in the model pattern. However, the similarity between the Petri nets and the model pattern should cross a threshold.

The definition of full evaluation is given as follows.

Definition 7 (Full Evaluation). Let PNS be a set of Petri nets, MP be a model pattern. Then the full evaluation of MP over PNS is defined as follows:

$$\begin{aligned} E_{full}(MP, PNS) = \{PN \mid & PN.P \supseteq MP.P, PN.T \supseteq MP.T \\ & , PN.F \supseteq MP.DF, PN.PS \supseteq MP.IF, PN \in PNS\} \end{aligned} \quad (1)$$

Example 1. There are two Petri nets in the Petri net repository as in Figure 4. A model pattern is put forward as in Figure 5. Both *Petri net*₁ and *Petri net*₂ contain the three vertices of the model pattern. In *Petri net*₁, there are paths from CHE to REV, one of which is (CHE, CHED, REV). But there is no path from REV to APR. According to the definition of full evaluation, *Petri net*₁ is not in the evaluation result. While paths from CHE to REV and from REV to APR can be found in *Petri net*₂, it is in the evaluation result.

In the case that too few Petri nets are got by full evaluation, we can relax the query by allowing Petri nets in the evaluation result only partially contain the information embedded in the model pattern. However, the similarity between the Petri nets and the model pattern should cross a threshold.

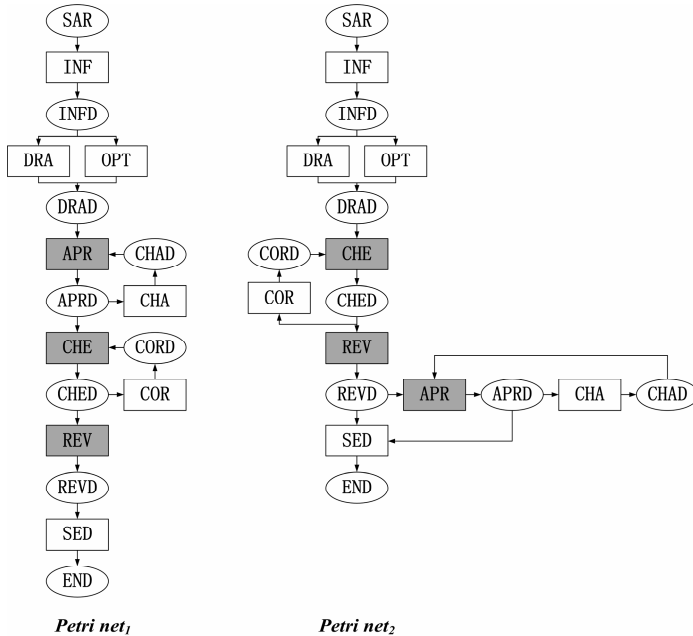


Fig. 4. Two Petri nets

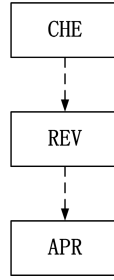


Fig. 5. Model pattern

The concept of similarity is fundamentally important in many scientific fields. Using set theory, Tversky [20] defined a similarity measure in terms of a matching process. This measure produces a similarity value that is the result of common as well as different characteristics of objects. This approach is in agreement with an information-theoretic definition of similarity [14]. The definition of Tversky's similarity is given as follows.

Definition 8 (*Tversky's Similarity*). Let A and B be two set, $A \cap B$ be the intersection of A and B , A / B be the difference of A and B , $| \cdot |$ be the cardinality of a set, and α be a function that defines the relative importance of the noncommon elements of two sets. The Tversky's similarity of A and B is defined as follows:

$$sim_{Tversky}(A, B) = \frac{|A \cap B|}{|A \cap B| + \alpha |A / B| + (1 - \alpha) |B / A|} \quad (2)$$

for $0 \leq \alpha \leq 1$.

The relative importance of the noncommon elements (shown in the second and third terms of the denominator on the right-hand side of equation (2)) allows the asymmetric evaluation of similarity.

The similarity between a Petri net and a model pattern can be evaluated from three aspects. The first aspect is the vertices aspect, the second aspect is the arcs aspect, and the third aspect is the paths aspect. Based on Tversky's similarity, their definitions are given as follows.

Definition 9 (*Vertices Similarity*). Let Petri net $PN = (P_1, T_1, F_1)$ be a Petri net and $MP = (P_2, T_2, DF_2, IF_2)$ be a model pattern. Then the vertices similarity is defined as follows:

$$sim_{vertex}(PN, MP) = \frac{|(P_1 \cup T_1) \cap (P_2 \cup T_2)|}{|(P_1 \cup T_1) \cap (P_2 \cup T_2)| + \alpha |(P_1 \cup T_1) / (P_2 \cup T_2)| + (1 - \alpha) |(P_2 \cup T_2) / (P_1 \cup T_1)|} \quad (3)$$

for $0 \leq \alpha \leq 1$.

Definition 10 (*Arcs Similarity*). Let Petri net $PN = (P_1, T_1, F_1)$ be a Petri net and $MP = (P_2, T_2, DF_2, IF_2)$ be a model pattern. Then the arcs similarity is defined as follows:

$$sim_{arc}(PN, MP) = \frac{|F_1 \cap DF_2|}{|F_1 \cap DF_2| + \alpha |F_1 / DF_2| + (1 - \alpha) |DF_2 / F_1|} \quad (4)$$

for $0 \leq \alpha \leq 1$.

Definition 11 (*Paths Similarity*). Let Petri net $PN=(P_1, T_1, F_1)$ be a Petri net and $MP=(P_2, T_2, DF_2, IF_2)$ be a model pattern. Then the paths similarity is defined as follows:

$$sim_{path}(PN, MP) = \frac{|PS_1 \cap IF_2|}{|PS_1 \cap IF_2| + \alpha |PS_1 / IF_2| + (1 - \alpha) |IF_2 / PS_1|} \quad (5)$$

for $0 \leq \alpha \leq 1$.

Based on the three similarity measures above, the synthesized similarity between a Petri net and a model pattern is given as follows.

Definition 12 (*Synthesized Similarity*). Let Petri net $PN=(P_1, T_1, F_1)$ be a Petri net, $MP=(P_2, T_2, DF_2, IF_2)$ be a model pattern, $sim_{vertex}(PN, MP)$, $sim_{arc}(PN, MP)$, and $sim_{path}(PN, MP)$ be the vertices, arcs, and paths similarity respectively, and w_{vertex} , w_{arc} , and w_{path} be their corresponding weights. Then the synthesized similarity is defined as follows:

$$sim_{synthesized}(PN, MP) = w_{vertex} \cdot sim_{vertex}(PN, MP) + w_{arc} \cdot sim_{arc}(PN, MP) + w_{path} \cdot sim_{path}(PN, MP) \quad (6)$$

for $w_{vertex}, w_{arc},$ and $w_{path} \geq 0$ and $w_{vertex} + w_{arc} + w_{path} = 1.0$

By default, the three aspects of similarity are considered equally important, i.e., $w_{vertex} = w_{arc} = w_{path} = 1/3$.

Equation (3), (4), (5) and (6) can be utilized to evaluate the similarity between a Petri net and a model pattern. As we are only concerned about the noncommon characteristics (i.e., vertices, arcs, and paths) of the model pattern, α in equation (3), (4), and (5) are set to be 0.

Based on the synthesized similarity, the definition of partial evaluation is given as follows.

Definition 13 (*Partial Evaluation*). Let PNS be a set of Petri nets, MP be a model pattern, τ be a threshold. Then the partial evaluation of MP over PNS is defined as follows:

$$\mathcal{E}_{partial}(MP, PNS, \tau) = \{PN \mid sim_{synthesized}(PN, MP) \geq \tau, PN \in PNS\} \quad (7)$$

Petri nets passing the partial evaluation do not have to fully contain the information embedded in the model pattern. Whereas, the synthesized similarity between the Petri nets and the model pattern should cross the threshold τ . In the case that too few Petri nets are got by full evaluation, the Petri nets with high similarity to the model pattern can be retrieved from the Petri net repository by partial evaluation. And the evaluation result can be ranked according to the synthesized similarity.

A Petri net passing the full evaluation fully contains the information embedded in the model pattern, thus the model pattern does not have noncommon characteristics. Therefore the synthesized similarity between the Petri net and the model pattern is equal to 1.

Furthermore, it is intuitive to have the fact that

$$\mathcal{E}_{full}(MP, PNS) = \mathcal{E}_{partial}(MP, PNS, 1) \quad (8)$$

Example 2. The Petri nets and the model pattern are the same as example 1. According to the definition of synthesized similarity, the similarity between the two Petri nets and the model pattern is given as follows.

$$sim_{synthesized}(Petri\ net_1, MP) = 5/6$$

$$sim_{synthesized}(Petri\ net_2, MP) = 1$$

If τ is set to be 0.8, according to the definition of partial evaluation, the two Petri nets are in the evaluation result.

5 Complexity of Evaluating Model Patterns

A fundamental issue in every query language is the complexity of query evaluation. In this section, we address the complexity of evaluating model patterns.

According to equation (8), full evaluation is a special case of partial evaluation, where the threshold τ is equal to 1. So we only study the complexity of partial evaluation. As it is customary when studying the complexity of the evaluation problem for a query language, we consider its associated decision problem. We define the problem as follows:

INPUT : A Petri net set PNS , a model pattern MP and a Petri net PN .

QUESTION : Is $PN \in \mathcal{E}_{partial}(MP, PNS, \tau)$?

Given a Petri net set PNS , a model pattern MP and a Petri net PN , it is possible to efficiently check whether $PN \in \mathcal{E}_{partial}(MP, PNS, \tau)$ by using the following algorithm.

Algorithm Partial Evaluation

procedure *partial_eval* (PNS : a Petri net set; MP : Model Pattern; PN : Petri net; τ : threshold)

for each place $p \in MP.P$

 search for p in $PN.P$

if $p \in PN.P$

then $p \in MP.P \cap PN.P$

else $p \in MP.P / PN.P$

for each transition $t \in MP.T$

 search for t in $PN.T$

```

if  $t \in PN.T$ 
    then  $t \in MP.T \cap PN.T$ 
    else  $t \in MP.T / PN.T$ 
for each direct arc  $df \in MP.DF$ 
    search for  $df$  in  $PN.F$ 
    if  $df \in PN.F$ 
        then  $df \in MP.DF \cap PN.F$ 
        else  $df \in MP.DF / PN.F$ 
for each direct arc  $if \in MP.IF$ 
    search for  $if$  in  $PN.PS$ 
    if  $if \in PN.PS$ 
        then  $if \in MP.IF \cap PN.PS$ 
        else  $if \in MP.IF / PN.PS$ 
calculate  $sim_{synthesized}(PN, MP)$  according to equation (3), (4), (5), and (6)
if  $sim_{synthesized}(PN, MP) \geq \tau$ 
    then return true
    else return false

```

The main operation of the algorithm is searching for an element in a set. In order to optimize the efficiency of search, set is stored as a binary search tree. The minimum height for a tree containing n nodes is $\lceil \log_2(n+1) \rceil$. Applying the algorithm in [21], given a sorted sequence of node values, a binary search tree of minimum height can be built in $O(n)$ time. And the search operation on a binary search tree takes time proportional to the height of the tree [5]. Vertices of Petri nets and model patterns can be sorted in alphabetical order according to their names. Arcs and paths can be sorted in alphabetical order according to the names of their initial and terminal vertex. Thus, the complexity of partial evaluation is

$$O(|MP.P| \cdot \lg |PN.P| + |MP.T| \cdot \lg |PN.T|, \\ + |MP.DF| \cdot \lg |PN.F| + |MP.IF| \cdot \lg |PN.PS|)$$

where $|\cdot|$ means the cardinality of the set.

Let n be the maximum of $|MP.P|$, $|PN.P|$, $|MP.T|$, $|PN.T|$, $|MP.DF|$, $|PN.F|$, $|MP.IF|$, and $|PN.PS|$. Then the complexity of partial evaluation is $O(n \cdot \lg n)$.

6 Conclusions and Future Perspectives

This paper develops PNQL, a graphical query language for Petri nets. PNQL is based on a similar graph-based view of Petri nets, which is called model pattern. The semantics

of model pattern is defined as an evaluation over a set of Petri nets. Full evaluation requires that every Petri net in the evaluation result fully contains the information embedded in the model pattern. In the case that too few Petri nets are got by full evaluation, the Petri nets with high similarity to the model pattern can be retrieved from the Petri net repository by partial evaluation. Partial evaluation allows the Petri nets in the evaluation result only partially contain the information embedded in the model pattern. Whereas, the synthesized similarity between the Petri nets and the model pattern should cross the threshold τ . And the evaluation result can be ranked according to the synthesized similarity. The complexity of evaluating model patterns is studied.

Semantic annotations can provide lots of valuable information about business process models. In recent years, rapid progress has been made in ontology query techniques. How to combine models' inherent structure and semantic annotations to support business process model query is one of our future research topics.

In our query mechanism, names of transitions (or places) in the model pattern and the query result, i.e., Petri nets, should be perfectly matched. However, equivalent transitions (or places) may be named differently by different users. How to match equivalent transitions (or places) with different names will be elaborated in our future works as well.

Acknowledgement

This research is partially supported by the National High Technology Research and Development Program of China (No. 2008AA04Z102).

References

1. Bowers, S., Ludascher, B.: A calculus for propagating semantic annotations through scientific workflow queries. In: *Proceedings of Query Languages and Query Processing Workshop*, pp. 712–723 (2006)
2. Braga, D., Campi, A., Ceri, S.: XQBE (xquery by example): A visual interface to the standard xml query language. *ACM transactions on database systems* 30(2), 398–443 (2005)
3. Cao, J., Zhao, H.: A semantic model and query mechanism for service workflow. In: *Proceedings of 2008 IEEE International Conference on Networking, Sensing and Control*, pp. 1850–1855 (2008)
4. Comai, S., Damiani, E., Fraternali, P.: Computing graphical queries over xml data. *ACM Transactions on Information Systems* 19(4), 371–430 (2001)
5. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: *Introduction to Algorithms*. The MIT Press, Cambridge (2001)
6. Dussart, A., Aubert, B.A., Patry, M.: An evaluation of inter-organizational workflow modeling formalisms. *Journal of Database Management* 15(2), 74–104 (2004)
7. Ellis, C.A., Nutt, G.J.: Workflow: The process spectrum. In: *NSF Workshop on Workflow and Process Automation in Information Systems*, pp. 140–145 (1996)
8. Goud, R., Van Hee, K.M., Post, R.D.J., Van Der Werf, J.M.E.M.: Petriweb: A repository for Petri nets. In: Donatelli, S., Thiagarajan, P.S. (eds.) *ICATPN 2006*. LNCS, vol. 4024, pp. 411–420. Springer, Heidelberg (2006)

9. Juan, Y.C.: A string comparison approach to process logic differences between business process models. In: *Proceedings of the 9th Joint Conference on Information Sciences* (2006)
10. Juan, Y.C., Ou-Yang, C.: A process logic comparison approach to support business process benchmarking. *International Journal of Advanced Manufacturing Technology* 26(1-2), 191–210 (2005)
11. Keramopoulos, E., Pouyioutas, P., Ptohos, T.: The GOQL graphical query language. *International Journal of Computers and Applications* 24, 122–128 (2002)
12. Lin, D.: An Information-Theoretic Definition of Similarity. In: *Proc. Int'l Conf. Machine Learning, ICML 1998* (1998)
13. Markovic, I., Pereira, A.C.: Towards a formal framework for reuse in business process modeling. In: *Proceedings of the 2nd International Workshop on Advances in Semantics for Web services*, pp. 484–495 (2008)
14. Oberweis, A., Sanger, V.: Graphical query language for simulation runs. *Journal of Microcomputer Applications* 17 (1994)
15. Oberweis, A., Stucky, W., Weitz, W., Zimmermann, G.: *INCOME/WF - A Petri Net-Based Approach to Workflow Management*, Institute fur Wirtschaftsinformatik, Germany (1996)
16. Perez, J., Arenas, M., Gutierrez, C.: Semantics and complexity of SPARQL. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) *ISWC 2006. LNCS*, vol. 4273, pp. 30–43. Springer, Heidelberg (2006)
17. Rosen, K.H.: *Discrete Mathematics and its Applications*. McGraw-Hill, New York (2003)
18. Tversky, A.: Features of Similarity. *Psychological Rev.* 84, 327–352 (1977)
19. Wang, C.M.: *BASS: BPR Analogy Support System*, Master Thesis: National Taiwan University of Science and Technology (1999)
20. Wang, Y., Cao, J., Li, M.: Goal-driven semantic description and query for grid workflow. In: *Proceedings of 3rd International Conference on Semantics, Knowledge, and Grid*, pp. 598–599 (2007)
21. Vaucher, J.G.: Building Optimal Binary Search Trees from Sorted Values in $O(N)$ Time. In: Owe, O., Kroghdahl, S., Lyche, T. (eds.) *From Object-Orientation to Formal Methods*. LNCS, vol. 2635, pp. 376–388. Springer, Heidelberg (2004)
22. van der Aalst, W.M.P.: Three good reasons for using a Petri net-based workflow management system. In: *Proceedings of the International Working Conference on Information and Process Integration in Enterprises (IPIC 1996)*, pp. 179–201 (1996)
23. van der Aalst, W.M.P.: Verification of Workflow Nets. In: Azema, P., Balbo, G. (eds.) *ICATPN 1997. LNCS*, vol. 1248, pp. 407–426. Springer, Heidelberg (1997)
24. van der Aalst, W.M.P., Desel, J., Oberweis, A. (eds.): *Business Process Management*. LNCS, vol. 1806. Springer, Heidelberg (2000)

Model Checking by Generating Observers from an Interface Specification Between Components

Tetsuo Hasegawa¹ and Yoshiaki Fukazawa²

¹ Waseda Univ./Toshiba Corp., Shinjuku Tokyo/Kawasaki Kanagawa, Japan
`tetsuo3.hasegawa@toshiba.co.jp`

² Waseda Univ., Shinjuku Tokyo, Japan
`fukazawa@waseda.jp`

Abstract. In the field of embedded software systems where many kinds of systems must be developed in a short period of time and at low cost, model checking, which is one of the automatic design verification techniques, is expected to become easy for software designers to use. The difficulties of model checking include the describing of queries or observers as the system property to be verified, and the analyzing of a counterexample in order to find the cause of a fault. There are methods to solve these problems such as generating observers from ordinary software design formats describing system behavior rules, and comparing that behavior with a counterexample to locate a reason for the fault. In this paper, a method generating observers from a timing diagram that describes an interface specification between two components is proposed. The purpose is to make it possible for designers to describe queries of verification easily and also analyze counterexamples easily. In addition, the result of applying this method to a communication protocol application is reported.

Keywords: model checking, UML, timing diagram.

1 Introduction

Model checking is one of the automatic design verification techniques. It checks whether a given model of a target system's behavior satisfies a given property. It produces an execution trace called a counterexample if the behavior violates the property. A bug in the system design may be found by analyzing the counterexample. Recently, various model checking tools have been developed [1] and examples of their application to real industry systems have been reported [2,3,4,5].

For embedded systems, real-time properties are important and UPPAAL has been developed as a model checking tool to verify real-time properties [6]. In the embedded systems field, since product life cycles are becoming shorter, development must be executed quickly and at low cost. However, software has become increasingly complicated, making it impossible for human designers to achieve

Table 1. Classification of verification properties, making methods of formulas and difficulty of counterexample analysis

classification of property to be verified	example of property to be verified	example method for creating query	difficulty of analyzing of an counter example
Property about occurrence of some status (Occurrence patterns)	do not go into a certain status X (absence), reach to a certain status Y (Existence), always keep some status condition (Universality)	general-purpose query which dose not depends on certain application system is used	analyzing a process to the status is required
Property about a sequence of state (Order patterns)	Some status X is a necessary pre-condition for some status Y (Precedence), Some status X must be followed by some status Y (Response)	creating an observer from desired behavior sequence	analyzing is easier by comparing desired behavior with counter example

design reviews verifying all behavior patterns. So, there is a great need for a model checking tool that they can use easily.

Research has been done on modeling system behavior easily [78]. They proposed methods for translation from a state diagram which is familiar to embedded software designers, to a timed automaton. They treat extended state diagrams that can represent real-time properties and translate them into timed automata of UPPAAL.

But regarding utilization of model checking, there are two major issues: the difficulty of contriving a suitable system property to be verified and defining it as a query or an observer, and the difficulty of analyzing a counterexample in order to find the cause of violation of the property.

For modeling system behavior, there are some reports on research on translation from a state diagram, which is familiar to embedded software designers, to a timed automaton [78]. They treat extended state diagrams that can represent real-time properties and translate them into timed automata of UPPAAL.

For defining a property to be verified, kinds of properties and corresponding query patterns are reported [9]. Based on these categories of properties, Table 1 shows the typical approach of defining a query or an observer and the difficulty of analyzing a counterexample against the query or the observer. As an example of a property concerning occurrence of a given status during system execution, a property desirable for many systems such as “system should not be in a deadlock status” can be used for arbitrary systems generally. But analyzing of a counterexample may be difficult in this case. For example, in order to localize the reason for deadlock, it is necessary to trace step by step from top to bottom of the counterexample in many cases. This is because the property does not represent any process of reaching the deadlock status and also does not involve any information about expected behavior.

In another approach to create a query, a rule of a behavior sequence such as an interface specification between components is considered as a property to be verified and then an observer monitoring whether the system behavior keeps the

rule is defined. In this case, it may be expected to be easy to localize the reason for the fault by comparing the behavior sequence of the counterexample with the behavior sequence represented by the observer.

So, in this paper we propose a method to generate observers from a timing diagram describing an interface specification between two components. The objective is to enable an embedded software designer who lacks expertise in model checking techniques to define the property to be verified and analyze a counterexample easily. The target model checking tool is UPPAAL because it can treat timing properties that are important in embedded systems. It is assumed that a system model is already generated as timed automata of UPPAAL by some other proposed approach.

In research on generating a property to be verified from a desired behavior sequence, a method of creating queries or an observer from a sequence diagram representing a message sequence has been proposed [10,11]. We use a timing diagram because it can represent not only a message sequence but also status transitions and timing properties.

In Section 2, terms relating to a system model and a query of UPPAAL are defined. In Section 3, the specification format of the interface between two components as input of the proposed method is defined and the method of generating an observer automaton is described. In Section 4, an application to a communication system is described and its evaluation is reported.

2 A System Model and a Query of UPPAAL

Terms related to description and semantics of a system model and a query for UPPAAL are defined here. They are based on the definition in [6] and [12].

- A timed automaton: a finite-state machine extended with clock variables. All the clocks progress synchronously.
- A system model: it is assumed that a system consists of several components executing in parallel. Behavior of a component is represented by a timed automaton. So a system is modeled as a network of several timed automata.
- A location and a system state: a system state is defined by locations of all timed automata and values of all clocks.
- A state transition: There are the following four kinds of state transitions.
- UPPAAL state transition(1): time passing: This transition causes values of all clock variables to increase. Locations of all timed automata are not changed.
- UPPAAL state transition(2): a transition between locations by a single component: This transition is called a location transition here. A location transition occurs without time passing. An initialization of a clock variable can be attached to a location transition. Only one process can make a location transition at a time with the exception of the following case.
- UPPAAL state transition(3): a transition between locations with a message passing through a binary channel: a transition can be attached message

sending into a channel or message receiving from a channel. For a transition with a binary channel, both a sender component and a receiver component make a location transition at one time.

- UPPAAL state transition(4): a transition between locations with message passing through a broadcast channel: a transition with message sending into a broadcast channel can occur at any time, and it causes location transitions of all components that are waiting for a message from the channel at the time.
- Restrictions caused by a committed location: time cannot be passing if a current location of any component is a committed location and a location transition from a committed location occurs prior to other location transitions from a non-committed location.
- A query: a query is defined as simplified version of CTL, computational tree logic. It consists of path formulae and state formulae. State formulae describe individual states and they include evaluation of clock variables. Path formulae quantify over paths or traces of the model. For example, “ $A \Box \text{not } (p.l \text{ and clock} > 10)$ ” means that the component p does not reach the location l while a value of the clock variable “clock” is over 10 at any time of any execution pass. A combination of path formulae and state formulae cannot be nested.
- An observer: an observer is a time automaton monitoring state transitions of the system with no effect on the system behavior. It is incorporated in the system model. Timed automata of the original system model may be modified in order to notify their location transition to the observer.

3 Generation of Observers

Observer automata are generated from the interface specification between two components. In this section, the specification format of an interface as an input of the method, the modification method of timed automata of the system model, and the method of generating observer automata are explained. Also, adequacy of generated observers’ behavior is mentioned.

3.1 Specification Format of an Interface between Two Components

The following points are considered to be important in designing the specification format of an interface between two components.

- The format should be familiar to embedded software designers. Timing constraint can be represented naturally and not only message sequence but also location transitions sequence of a component can be described.
- An interface can be focused on only interaction between two components of interest. Also it should be necessary to describe only location transitions participating in the interaction.

A proposed specification format is based on a timing diagram defined in UML2.0. There are two reasons that a timing diagram is suitable. First, because it is one

kind of interaction diagram of UML, message dialogue between components can be defined. Second, it can represent real-time properties. Some limitations and extensions are introduced.

Each state in a timing diagram corresponds to a location in the timed automaton of the component. That is, a state transition sequence of a state timeline corresponds to a part of a location transition sequence of the component. A message sending event occurs only at a time of state transition, not during staying in a state. A message sending event is not described explicitly and it is identified by a transition between specified states. A state timeline has all the state transitions during the interval described by the state timeline. But in order to avoid describing state transitions irrelevant to the interface specification, a special state is introduced. It is called an arbitrary state. When a state timeline is located in an arbitrary state, it means any state transition and any message event may occur. That is, it corresponds to any sequence of state transitions in a timed automaton of the system model. The first state of a state timeline is an arbitrary state. This arbitrary state corresponds to a location transition sequence from initial state to the behavior of the specified interface.

Timing constraints are described by time observations and time constraints. A time observation is considered to be the starting point for measurement of time proceeding. Clock variables are assigned to each starting point and represented as substitution of current time to them such as “ $t=now$ ” in Figure 1. These clock variables are shared with two state time lines in the timing diagram. On the other hand, a time constraint indicates the minimum and the maximum value of the clock variable at that point. It is represented as “ $t2..t2+5$ ” in Figure 1. If constraint does not have the minimum or the maximum, the corresponding value is not inscribed.

Time observations and time constraints are located at a state transition. Each timing constraint is annotated as an “assumed” constraint or a “target” constraint. Interface specification is considered as that if system behavior has execution paths including the specified location transition sequence and all assumed time constraints are satisfied, all target time constraints must be satisfied.

Figure 1 shows an example of the interface specification. In the figure, states A1 and A2 correspond to locations A1 and A2 of a timed automaton for component A, respectively. Similarly, states B1 and B2 correspond to locations B1 and B2 of a timed automaton for component B, respectively. This diagram indicates the following behavior. If the case of component A and B behaves such that after component A transits into A1, component B transits into B1, then into B2 after any state transitions and then component A transits from A1 to A2, that transition from A1 to A2 of component A must occur 0 to 5 time units after the transition of component B into B2.

3.2 Modification of Timed Automata of a System Model

Timed automata of a system model are modified in order to notify their transition to observers.

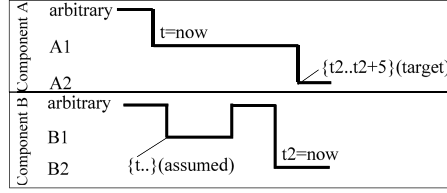


Fig. 1. An example of timing diagram as input

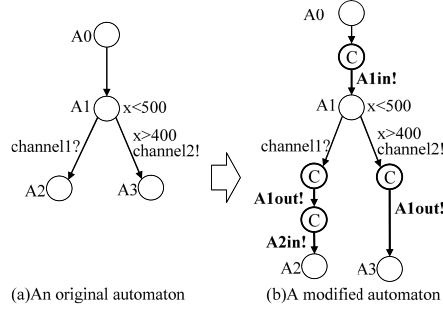


Fig. 2. An example modification of a timed automaton

Sending a message through a broadcast channel is used for notification. A sender does not block even if receiver does not exist. Modification is that the following two extra transitions attached by sending a message through a broadcast channel combined with committed locations are inserted after an existing transition corresponding to each transition described in the state timeline of the timing diagram. The labels of these broadcast channels mean a location transition from the previous location in the state timeline and a location transition to the subsequent location in the state timeline.

As an example, Figure 2(b) shows modification on a timed automaton Figure 2(a) if Figure 1 is given as an interface specification. In this example, because A1 and A2 are included in the timing diagram, the timed automaton is modified as follows. First a transition attached message sent through a broadcast channel named A1in is inserted to the transition into A1. Similarly, a transition with a message sent through A1out is inserted to both transitions from A1. For the transition into A2, transition with a message sent through A2in is inserted, but for the transition into A3, no transition is inserted because the timing diagram does not include A3.

3.3 Procedure of Generating Observers

Firstly an observer is generated by referring only to each component's state timeline. Then both observers are modified to be reflected in time constraints that deal with both components. A procedure for generating an observer for each state timeline is as follows:

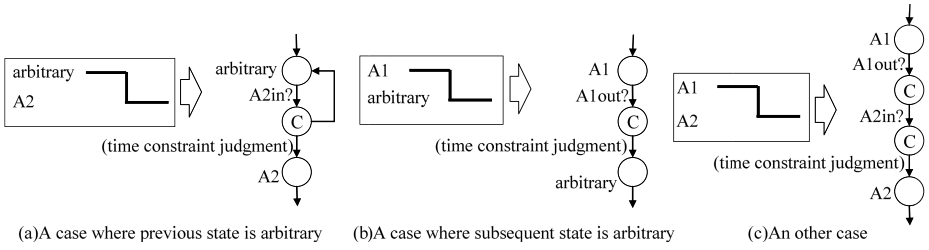


Fig. 3. Element units of an observer

1. Declare clock variables corresponding to time observations and a flag variable "failflag" that becomes "1" when any time constraint is violated.
2. Define an initial location corresponding to the first arbitrary state of the state timeline.
3. Append an element unit of an observer described later for all state transitions (not states) in a state timeline.
4. Finally, append locations named "finish" and "fail" after the last location. A transition to "finish" location is guarded by "failflag==0" and another transition to "fail" location is guarded by "failflag!=0".

There are the following three types of element units of an observer.

- For the case that a previous state is arbitrary: An element unit consists of two locations combined with a committed location and two transitions. The first location corresponds to the arbitrary state and another denotes the subsequent state in the state timeline. The first transition is attached to a message receiving through a channel whose label means transition into a subsequent state. The second transition has a judgment process for a time constraint as described later. Also, a transition from the middle committed location to a previous arbitrary location is appended. (Figure 3(a))
- For the case that a subsequent state is arbitrary: It consists of two locations combined with a committed location and two transitions. The first location corresponds to the precedent state in the state time line and the next one denotes the arbitrary state. The first transition is attached to a message receiving through a channel whose label means a transition from a previous state. The second transition has a judgment process for time constraint. (Figure 3(b))
- For other case (hereafter we call it the usual case): It consists of two locations combined with two committed locations and three transitions. The first location corresponds to the precedent state and the next one denotes the subsequent state in the state timeline. The first transition is attached to a message receiving through a channel whose label means a transition from a previous state. The second transition is attached to a message receiving through a channel whose label means a transition into a subsequent state. The last transition has judgment process for time constraints. (Figure 3(c))

As the judgment process for a time constraint, if a time constraint and/or a time observation are declared at the transition of a state timeline, a guard with the time constraint and/or an initialization of the clock variable corresponding

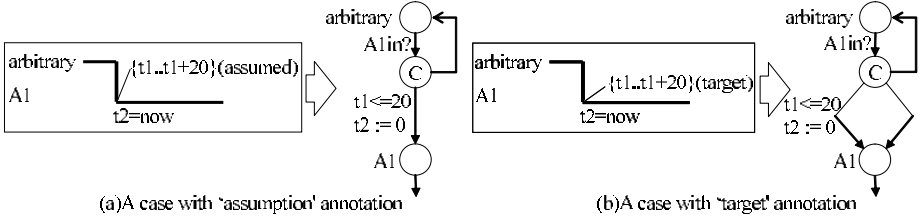


Fig. 4. Examples of time constraint judgement

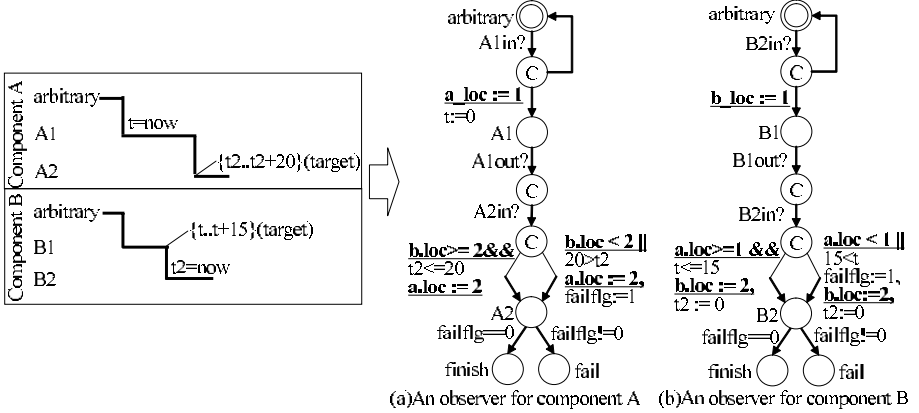


Fig. 5. Example observers for two components

to them is assigned (Figure 4(a)). If the time constraint is annotated with “target”, one more transition is appended. It has a guard with negation of the time constraint and an update “failflag” to 1. Also it has an initialization of the clock variable corresponding to the time observation if it is declared (Figure 4(b)).

Query expression is fixed as “ $A \square \text{not observer.fail}$ ”. If a system behavior is inconsistent with timing constraints, observers will transit into a location labeled “fail”. Then by this query expression, a counterexample will be created.

After observers are generated for both components by the above procedure, in the next step both observers will be modified according to the following procedure. Variables representing the current location of each observer are declared. Each variable is updated to a value indicating the next state at each transition. Then, for each judgment of time constraint, if its guard has a time constraint, an additional condition indicating whether the clock variable referred to in the constraint has already been initialized for starting measurement is added.

An example generated according to this procedure is shown in Figure 5. Underlined parts of the figure are added by this procedure.

3.4 Adequacy of a Generated Observer's Behavior

It is necessary to confirm the following two points in order to prove the adequacy of the proposed method.

1. Modification of timed automata does not have any influence on its behavior.
2. For all location transition sequences corresponding to the specified timing diagram (called assumption sequence here) that appeared in all execution paths of the component's behavior, the observer must make corresponding transitions and must reach its finish or fail location.

Concerning the first point, modification of timed automata of a system model involves appending the following transitions combined by a committed location. Appended transitions are attached only to messages sent through a broadcast channel. According to the UPPAAL state transition(4) mentioned in Section 2, no waiting occurs regardless of whether receivers are already waiting. Also, these transitions occur prior to other ones according to the restrictions caused by a committed location. So, it is obvious that the modification has no effect.

Concerning the second point, UPPAAL searches for all execution paths. But according to the UPPAAL state transition(4), a transition with message receiving from a broadcast channel must be transited when some component transits with a message sending to that channel, and so an observer is enforced to make transitions for first assumption sequence in each execution path. The extra transition from middle committed location to previous arbitrary location in element units of an observer solves this problem. It provides other behaviors of the observer that correspond to assumption sequences left in each execution path.

4 Application to a Communication Protocol System and Evaluations

We applied the proposed method to a communication protocol system among items of audio equipment reported as a case study of UPPAAL [5]. This system consists of multiple senders and a receiver connected via a wire. Bit sequence of transferred data is represented by voltage change timing of the wire.

According to this protocol, a case that multiple senders send bit sequences at the same time they work exhibits the following behavior. While they send the same bit sequence, they will continue. Sender can recognize a collision if actual voltage of the wire is high even if it tries to set it low. Then that sender stops data sending.

A sender must change the voltage of wire at appropriate timing according to the protocol. So this application can be considered to be one of the real-time applications. In the paper [5], it is verified that the protocol allowed a timing uncertainty of 5%.

Besides senders and a receiver, a system model consists of a wire and a sending data generator, a checker that gets bit data from the generator and from the receiver and then compares them. A property to be verified in the paper [5] is that the checker always receives the same bit data from both the generator and

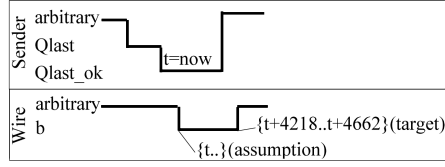


Fig. 6. An input timing diagram

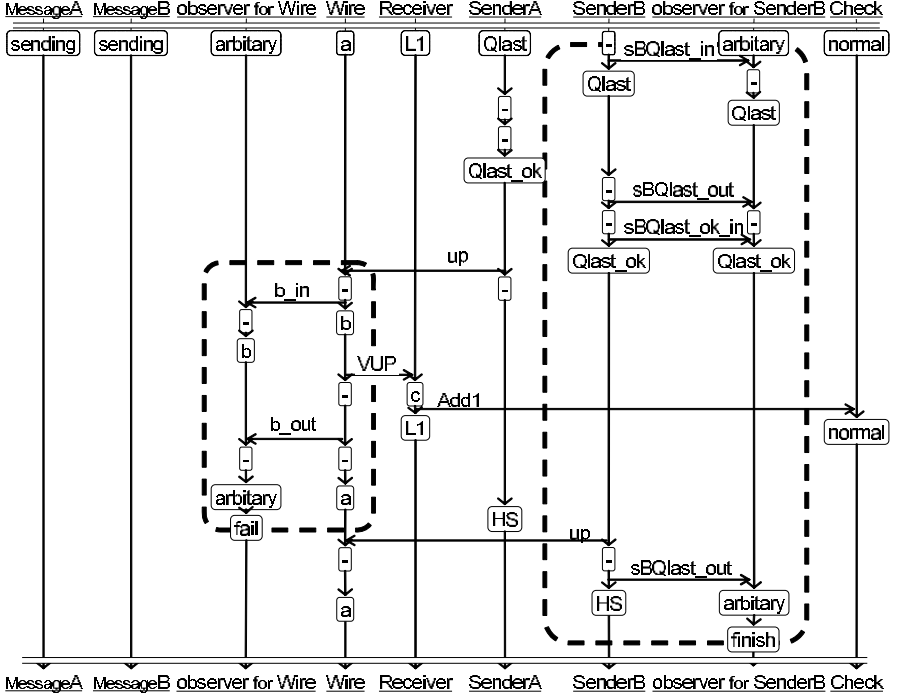


Fig. 7. A part of counterexample corresponding to a timing diagram

receiver. A counterexample includes whole steps from when the generator creates the first bit till when the receiver recognizes the sent bit incorrectly. In order to understand the reason for this fault, it is necessary to trace the counterexample from top to bottom by checking when each sender tries to change voltage of the wire and when voltage of the wire is changed.

The proposed method is applied to this system. An interface between a sender and a wire for a case that a bit "1" is sent is given. A property is that a timing uncertainty of time period from when a sender checks a collision until when voltage of a wire is actually increased must be under 5%. The timing diagram representing this specification is shown in Figure 6. When a sender finishes checking a collision, it transits into a location "Qlast_ok". The sender requests the wire to increase the voltage when it transits from a location "Qlast_ok". Voltage of wire

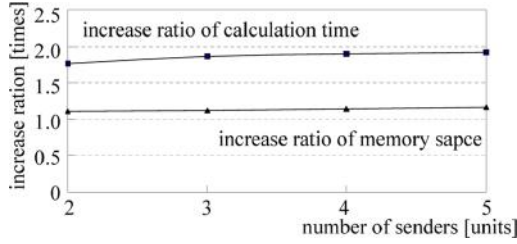


Fig. 8. Overhead by modification of components' timed automata

becomes high when a wire transits from a location “b”. A time passing between these two transitions must be in the time range 4218 to 4662.

In each counterexample, behavior of observers is straightforward and they indicate just the steps corresponding to the specified interface that is violated in many steps of the counterexample. Figure 7 shows a part of a counterexample where observers make transitions.

Location transitions of “observer_for_SenderB” and “observer_for_Wire” represent location transitions of “sender_B” and “Wire”, respectively, regarding the interface between them. These transitions are surrounded with dot-line in Figure 7. They corresponds to the range of counterexample from the 389th to the 408th transition. From the first transition to the shown part, both observers located at “arbitrary” location. So it is easy to focus on the part violating the interface.

According to the counterexample, there is no message exchange between “Sender_B” and “Wire”. “Wire” is triggered by message from “Sender_A”. As mentioned above, it is easy to compare the expected behavior between “Sender_B” and “Wire” with actual behavior.

It is confirmed that a counterexample can be analyzed easily in view of the following points.

- The timing diagram specified as an interface between the sender and the wire components shows expected behavior of these components.
- So each location transition in the counterexample corresponds to a state transition in the timing diagram.
- In the result, the part of the counterexample concerning the behavior of components violating the specified interface can be focused on and also distinguished from interleaved steps unconcerned with the interface.

Next, the overhead of this method is evaluated. The proposed method appends extra locations and broadcast channels as modifications for original components' timed automata. Concretely numbers of locations increase from 12 to 20 for “sender” and from 2 to 4 for “wire” and 10 broadcast channels are added. Generally, increase of locations and channels are significant influence to search area of a model checking and they may make large overhead.

In order to measure the overhead, the rate of increase is calculated by comparing memory space and calculation time for the original timed automata with

those for modified timed automata. A tool named “memtime” that is provided with the UPPAAL tool is used. In this application example, number of “sender” component can be changed; it is varied 2 to 5 for searching an overhead for larger system model. Figure 8 shows a result. Memory space increases 1.1 times to 1.2 times and calculation time increases 1.8 times to 1.9 times. An overhead is not negligible but it does not increase extremely even if original size of system model is increased. Only committed locations that are suppressive of search space increase are appended, which may help to prevent search space explosion. Since the calculation power of computers will increase in the future, this non-exponential overhead can be considered to be a practical level.

5 Conclusion

This paper proposes a method of generating observers for UPPAAL automatically from an interface specification between two components in UML’s timing diagram as a verification property. The purpose is to resolve difficulties of model checking for software designers who are not experts in model checking. The proposed method is evaluated with an example application.

References

1. Berard, B., Bidoit, M., Finkel, A., Laroussinie, F., Petit, A., Petrucci, L., Schnoebelen, P., Mckenzie, P.: *Systems and Software Verification: Model-Checking Techniques and Tools*. Springer, Heidelberg (2001)
2. Havelund, K., Lowry, M., Penix, J.: Formal analysis of a space craft controller using Spin. In: *Proc. of 4th International SPIN Workshop* (1998)
3. Janssen, W., Mateescu, R., Mauw, S., Springintveld, J.: Verifying business processes using SPIN. In: *Proc. of 4th International SPIN Workshop* (1998)
4. Lindahl, M., Pettersson, P., Yi, W.: Formal Design and Analysis of a Gear-Box Controller. In: Steffen, B. (ed.) *TACAS 1998*. LNCS, vol. 1384, p. 281. Springer, Heidelberg (1998)
5. Bengtsson, J., Griffioen, W.O.D., Kristoffersen, K.J., Larsen, K.G., Larsson, F., Pettersson, P., Yi, W.: Verification of an Audio Protocol with Bus Collision Using Uppaal. In: Alur, R., Henzinger, T.A. (eds.) *CAV 1996*. LNCS, vol. 1102. Springer, Heidelberg (1996)
6. Behrmann, G., David, A., Larsen, K.G.: A Tutorial on Uppaal. In: Bernardo, M., Corradini, F. (eds.) *SFM-RT 2004*. LNCS, vol. 3185, pp. 200–236. Springer, Heidelberg (2004)
7. David, A., Moller, M.O., Yi, W.: Formal Verification of UML Statecharts with Real-Time Extensions. In: Kutsche, R.-D., Weber, H. (eds.) *FASE 2002*. LNCS, vol. 2306, p. 218. Springer, Heidelberg (2002)
8. David, A., Moller, M.O.: From HUPPAAL to UPPAAL: Translation from hierarchical timed automata to flat timed automata, BRICS Technical report series, RS-01-11 (2001)

9. Dwyer, M.B., Avrunin, G.S, Corbett, J.C.: Patterns in Property Specifications for Finite-State Verification. In: Proceedings of the 21st International Conference on Software Engineering (May 1999)
10. Inverardi, P., Muccini, H., Pelliccione, P.: Automated Check of Architectural Models Consistency Using SPIN. In: Proc. of 16th ASE 2001(2001)
11. Firley, T., Huhn, M., Diethers, K., Gehrke, T., Goltz, U.: Timed Sequence Diagrams and Tool-Based Analysis A Case Study. In: France, R.B., Rumpe, B. (eds.) UML 1999. LNCS, vol. 1723, pp. 645–660. Springer, Heidelberg (1999)
12. Bengtsson, J., Wang, Y.: Timed automata: Semantics, algorithms and tools. In: Desel, J., Reisig, W., Rozenberg, G. (eds.) Lectures on Concurrency and Petri Nets. LNCS, vol. 3098, pp. 87–124. Springer, Heidelberg (2004)

A Meta-modeling Framework to Support Accountability in Business Process Modeling

Joe Zou^{1,2}, Christopher De Vaney¹, and Yan Wang²

¹ IBM Australia

² Macquarie University, Sydney, NSW, Australia

{joezou,cdevaney}@au1.ibm.com, yanwang@ics.mq.edu.au

Abstract. Accountability is becoming a central theme in business today in the midst of global financial crisis as the corporate scandals and fallouts dominate the front pages of the press. Businesses are demanding more accountability measures built-in at the business process modeling level. Currently the business process modeling standards and methods mainly focus on the sequential flow aspect of business process and leave the business aspect of accountability largely untouched. In this paper, we extend the OMG's business modeling specifications to define a business accountability meta-model. The meta-model is complementary to the OMG's Model-Driven Architecture (MDA) vision, laying out the foundation for future model generation and transformation for creating accountable business process solutions.

Keywords: Accountability, Business Process Modeling.

1 Introduction

Business Process Management (BPM) has been widely adopted by today's enterprises to streamline their business processes and ultimately optimize their business performance. An important aspect of BPM is business process modeling, which uses tools and methods built on robust business process meta-models to enable process analysis, design and simulation. Currently the Object Management Group (OMG) is working on a series business modeling specifications such as Business Motivation Model (BMM) [1], Semantics of Business Vocabulary and Business Rules (SBVR) [2], Business Process Definition Meta-model (BPDm) [3], Business Process Modeling Notation (BPMN) [4], Organization Structure Meta-model (OSM) [5] and Production Rules Representation (PRR) [6] to support the BPM initiative.

Recently, the issue of accountability has become a major concern for businesses around the world in aftermath of a lot of corporate scandals and fallouts. The most significant examples are the Ponzi scheme conducted by the high profile funds management group in US and the contamination of dairy products from institutions in China. Businesses are keen to find ways to evaluate and improve accountability in business processes, especially in the Business Process Outsourcing (BPO) area. Traditionally, accountability is implicitly touched on in the IT governance processes and best practices such as Cobit and ITIL, but tools and methods are nonetheless lacking for businesses to clearly model accountability requirements at the business process

level. This paper defines an accountability meta-model based on the OMG's business modeling specifications. The accountability meta-model can then be used by businesses to model accountability requirements in business processes at both the Computation Independent Model (CIM) and the Platform Independent Model (PIM) levels, laying out the foundation work for future model transformation from those levels to implementation level models, i.e. Platform Specific Model (PSM).

The reminder of the paper is organized as follow. In the next section, we review the existing literature on accountability modeling. Then in section 3 we discuss how the accountability meta-model fits in the OMG's Model-Driven Architecture vision. In section 4 we extend the OMG business modeling specifications to define a business accountability meta-model and formally specify the key model constructs. This is followed in section 5 on how to apply the model to address the accountability issue during the lifecycle of a service contract. Finally we conclude this paper, outline its contributions and discuss further work.

2 Related Work

Traditionally, accountability has been extensively researched in the management literature. Oakes and Young note that the accountability concept is broad and difficult to characterize precisely [7]. They find while some researchers define accountability as "answerability" or an "obligation to account for how well resources used to meet specified outcomes", others argue that accountability is either "credit or blame" or "corporate scapegoating" [7]. A broadly accepted definition available at Wikipedia refers Schedler, who defines accountability as the "obligation to inform other parties about actions and decisions, or justify them and to be punished in the case of misconduct" [8]. Gibbins and Newton suggest that the obligation normally stems from a relationship driven by social, contractual, hierarchical or other factors [9]. The accountability concept in IT literature has several different meanings including non-repudiation [12, 13, 14]; ownership of the responsibility to meet requirements in end-to-end process [15]; root cause detection, diagnosis and defusing [16]; and being "answerable" and "explainable" in information and communication technologies [17]. The diversity of accountability definitions presents challenges to accountability requirement modeling in IT solution. In [10], the authors unify accountability definitions in management and IT literature by defining accountability as "obligations" for execution and fulfilment of a service. The obligations include "obtaining trusted agreement", "answering, providing explanation", "full disclosure" and "assuming undeniable liability for results". However, a precise definition of accountability meta-model is still missing in the literature. While some meta-models exist focusing on business activity monitoring (BAM) in BPM, they are narrowly emphasizing on either measuring or monitoring aspect of accountability. In [17], the authors define a business process meta-model using UML 2 profile. The meta-model includes accountability concepts likes *Measure*, *Process Goal*, *Deliverable* and *Process Owner*, but leaves the monitoring aspect largely unattained. An MDA approach to business monitoring is presented in [18], which defines a meta-model for process performance indicator (PPI) monitoring and outlines a model transformation technique for creating PSM process models that encompasses monitoring. While the technique is promising, the model currently only addresses the monitoring and measuring aspects of accountability.

In summary, a precise definition of accountability meta-model is missing in the literature. Given the increasing important nature of the accountability concern in the current climate, it is crucial to have an accountability meta-model formally specified using open standards, such as the OMG's business modeling specifications.

3 Accountability Meta-model in OMG MDA Framework

The OMG's vision is to build a Model-Driven Architecture framework that enables IT practitioners to build adaptive IT solutions that separate business and application logic from the underlying platform technology [19]. The MDA approach helps modelers to model requirements and solutions at various abstraction levels, starting from CIM, to PIM and PSM. It also allows semi-automated or automated mechanisms for transforming models from abstract to more specific level. Moreover, it facilitates code generations from PSM. At the business model layer, currently OMG provides SBVR, BPDM, BPMN, OSM, BMM and PRR specifications. SBVR defines the meta-model for business rules and vocabularies [2]; BPDM and BPMN present a business process meta-model and process modeling notation [3, 4]; OSM specifies the organization structure meta-model [5]; BMM introduces a meta-model for business motivations [1] whereas PRR provides a standard production rule representation for business rules modeling [6]. While these specifications lay out a foundation for business modeling, the crucial accountability concern in business process is not covered. This paper builds a business accountability meta-model based on the OMG's business model specifications to facilitate an MDA approach in solutions that address accountability requirements such as information disclosure and Business Activity Monitoring (BAM). Fig. 1 illustrates how the proposed accountability meta-model positions in the current MDA framework from OMG [2].

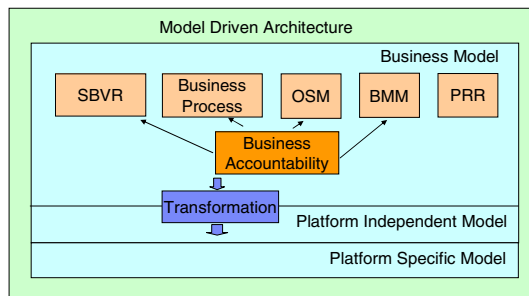


Fig. 1. Business Accountability in OMG MDA

4 Accountability Meta-model

4.1 Service Contract Meta-model

As accountability is closely related to an established contract, we first define a Service Contract meta-model based on the OMG's business modeling specifications.

Outsourcing (BPO) situation. As illustrated in Fig. 3, the *AccountableParty* has *Accountability* on a *BusinessProcess*. The *BusinessProcess* accepts *BusinessItem* as input and produces a set of *Deliverable*. The *BusinessProcess* has a defined *ProcessGoal* to achieve and also involves *Risk* that may undermine the achieving of the goal. The *BusinessProcess* consists of a series of *BusinessTask*, which creates various *ProcessSituation*. The *AccountableParty*'s *Accountability* is to mitigate the *Risk* through a series of tasks including disclosing *DeliverableSpecification* and *InputSpecification*; monitoring the *ProcessSituation*; proceducing *EventData*; raising possible *Alert*; logging *Evidence*; disclosing *ProcessStatus*; monitoring *SLA*; measuring *Metric* and *BusinessMeasure*; reporting *KPI*, complies to *RegulatoryProvision* and finally honoring *Remediation* if the service contract is breached by the *AccountableParty*. All of the classes in the model extend the existing concepts from OMG's SBVR, OSM and BMM. For example, *Accountability* extends the *ObligationStatement* from SBVR, *AccountableParty* extends the *LegalEntity* from OSM, and *ProcessGoal* extends *Goal* from BMM, and so on so forth.

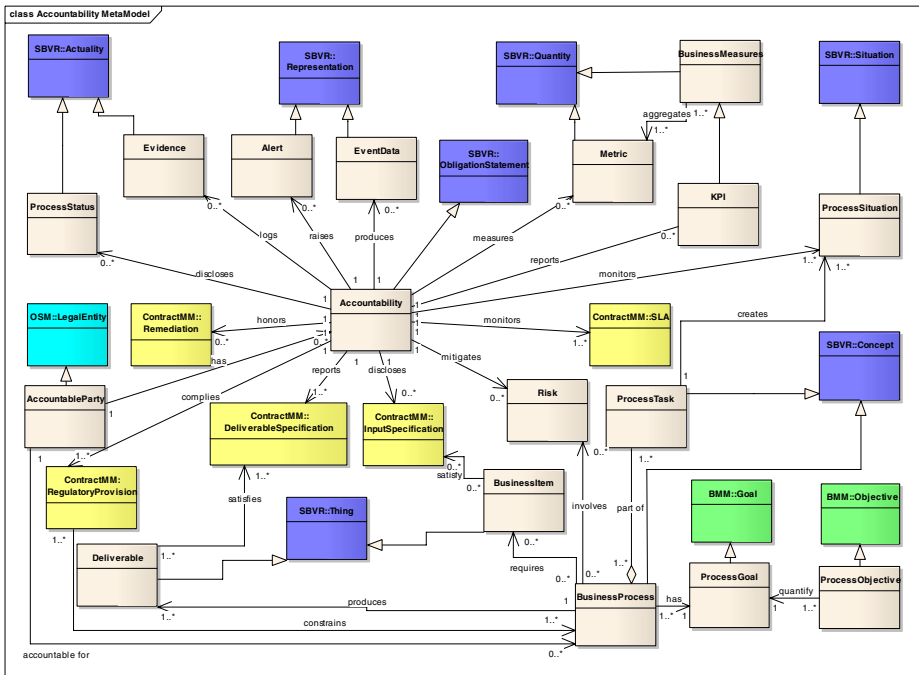


Fig. 3. Definition of an Accountability Meta-model

4.3 Formal Specification of the Contract and Accountability Meta-model

Assume that service provider Sp offers a service S , which is a business process that takes a set of business items $Bitem = \{b_1, b_2, \dots, b_n\}$ as input, and produces a set of deliverables $D = \{d_1, d_2, \dots, d_n\}$. Also assume that service consumer Sc wishes to

consume the service S . Sc negotiates with Sp and forms a contract C , which binds two contracted parties with a set of deliverable specifications (type *Dspec-set*), a set of input specifications (type *Ispec-set*), a service-level agreement (type *SLA*), a regulatory provision document, a remediation document (type *Remediation*) that specifies the penalty if the specification is not met, and the valid time duration for the contract. In this context the specification is a set of assertions or constraints that the specified object must satisfy. Accordingly, we can use VDM[11] to specify contract C as: (For VDM specific keywords or symbols, see Appendix A.)

$C::$ *provider*: LegalEntity
 consumer: LegalEntity
 po: ProviderObligation
 co: ConsumerObligation
 rg: RegulatoryProvision
 remedy: Remediation
 validFrom: Date
 validTill: Date

where

$\text{inv } (\mathbf{mk}\text{-}C(p, c, po, co, rg, rd, vf, vt)) \triangleq$
 $\forall p \in \text{LegalEntity} \bullet \forall c \in \text{LegalEntity} \bullet (p \neq c \wedge (\forall vf, vt \in \text{Date} \bullet vf \leq vt))$

ProviderObligation is the provider's obligations under the contract, which can be specified as:

ProviderObligation:: *deliverableSpec*: *Dspec-set*
 serviceLevel: *SLA*

ConsumerObligation is the consumer's obligations under the contract, which can be specified as:

ConsumerObligation:: *inputSpec*: *Ispec-set*
 c: *Consideration*

where

$\text{inv}\text{-}ConsumerObligation(\mathbf{mk}\text{-}ConsumerObligation(i, c)) \triangleq (c > 0)$

The service S under the contract C can be specified as:

$S(Bi: \text{Bitem-set}) \text{ De: } D\text{-set}$
 ext rd $c: C$
 pre $\text{assert-input}(Bi, \text{inputSpec}(co(c)))$
 post $\text{assert-output}(De, \text{deliverableSpec}(po(c)))$

Basically service S needs to ensure that all of the deliverables satisfy the deliverable specifications in contract C if the input items satisfy the input specifications in C . The function *assert-input* asserts that the input business items satisfy the business item specifications. It assumes that there is a one-to-one map between the business item set and the input specification set.

assert-input(*Bi*: **Bitem-set**, *Ii*: **Ispec-set**) *b*: **B**
 pre $\exists m \in (Bi \leftrightarrow^m Ii) \bullet m \neq \{ \}$
 post $b = \forall bi \in Bi \bullet \exists m \in (Bi \leftrightarrow^m Ii) \bullet m \neq \{ \} \wedge i\text{-satisfies}(bi, m(bi))$

where *i-satisfies* is a function to check if a particular item satisfies the corresponding specification. Its function signature is:

i-satisfies: $Bitem \times Ispec \rightarrow \mathbf{B}$

Similarly, *assert-output* can be specified as:

assert-output(*De*: **D-set**, *Ds*: **Dspec-set**) *b*: **B**
 pre $\exists m \in (De \leftrightarrow^m Ds) \bullet m \neq \{ \}$
 post $b = \forall d \in De \bullet \exists m \in (De \leftrightarrow^m Ds) \bullet m \neq \{ \} \wedge d\text{-satisfies}(d, m(d))$

where

d-satisfies is a function to check if a particular deliverable satisfies the corresponding specification. Its function signature is:

d-satisfies: $D \times Dspec \rightarrow \mathbf{B}$

With contract *C* and service *S* specified, now we can specify the key accountability functions in Fig. 4 for service provider *Sp*. Accountability can be modeled as operation requirements during the life cycle of the service contract.

- 1) Disclosing obligations in the contract *C*:

DISCLOSE_OBLG() *d*: *Document*
 ext rd *c*: *C*
 pre $c \neq \text{nil}$
 post $d = \text{formatOblg}(\text{deliverableSpec}(po(c)), \text{serviceLevel}(po(c)))$

where

formatOblg is a function to serialize the deliverable specifications and SLA into a document. Its function signature is:

formatOblg: $Dspec\text{-set} \times SLA \rightarrow Document$

- 2) Monitoring service to record event data and raise alert if the event is abnormal:

MONITOR_SER
 ext rd *ps*: *ProcessSituation*, wrt *e*: *EventData-set*, wrt *a*: *Alert-set*
 post $e = (\{ \text{getEvent}(ps) \} \cup \tilde{e}) \wedge (\text{abnormalEvent}(\text{getEvent}(ps)) \Rightarrow a = (\{ \text{getAlert}(\text{getEvent}(ps)) \} \cup \tilde{a}))$

where

getEvent is a function to extract event data from a process situation; *abnormalEvent* is a function to check if the event is abnormal; and *getAlert* is a function to map abnormal event to alert. Their function signatures are:

getEvent: *ProcessSituation* $\rightarrow$ *EventData*

abnormalEvent: *EventData* $\rightarrow$ **B**

getAlert: *EventData* $\rightarrow$ *Alert*

3) Measuring metrics, business measures and KPIs:

MEASURE_METRIC_KPI

ext rd *e*: *EventData-set*, wrt *m*: *Metric-set*, wrt *bm*: *BusinessMeasure-set*,

wrt *k*: *KPI-set*

pre $e \neq \{\}$

post $m = \text{calMetric}(e) \wedge (m \neq \text{nil} \Rightarrow (bm = \text{aggBMea}(m) \wedge (k = \text{getKPI}(bm))))$

where

calMetric is a function to calculate metrics from event data; *aggMeas* is a function to aggregate metrics to business measures whereas *getKPI* is a function to get KPIs from the business measures. Their function signatures are:

calMetric: *EventData-set* $\rightarrow$ *Metric-set*

aggBMea: *Metric-set* $\rightarrow$ *BusinessMeasure-set*

getKPI: *BusinessMeasure-set* $\rightarrow$ *KPI-set*

4) Reporting KPI:

REPORT_KPI(*d*): *Document*

ext rd *k*: *KPI-set*

pre $k \neq \{\}$

post $d = \text{formatKPI}(k)$

where

formatKPI is a function to serialize KPI data to a document. Its function signature is:

formatKPI: *KPI-set* $\rightarrow$ *Document*

5) Disclosing process status via answering service consumer's inquiry:

DISCLOSING_STATUS(*q*: *Query*) *d*: *Document*

ext rd *e*: *EventData-set*, wrt *s*: *ProcessStatus*

pre $e \neq \{\} \wedge q \neq \text{nil}$

post $s = \text{getStatus}(e) \wedge d = \text{formatStatus}(s)$

where

getStatus is a function to work out the process status based on the event data store and *formatStatus* serializes process status data to a document. Their function signatures are:

getStatus: *EventData-set* $\rightarrow$ *ProcessStatus*

formatStatus: *ProcessStatus* $\rightarrow$ *Document*

6) Logging Evidences:

LOG_EVIDENCE

ext rd *ps*: *ProcessSituation*, wrt *e*: *Evidence-set*

post $e = (\{ \text{getEvidence}(ps) \} \cup e^c)$

where

getEvidence is a function to extract evidence from a process situation. Its function signature is:

getEvidence: *ProcessSituation* $\rightarrow$ *Evidence*

7) Honoring Remediation:

HONOR_REM ()

ext rd *c*: *C*, rd *Bi*: *Bitem-set*, rd *De*: *D-set*, rd *K*: *KPI-set*

pre $c \neq \text{nil} \wedge \text{assert-input}(Bi, \text{inputSpec}(co(c))) \wedge (\neg \text{assert-output}(De, \text{deliverableSpec}(po(c))) \vee \neg \text{assert-sla}(K, \text{serviceLevel}(po(c))))$

post *compensate*(*remedy*(*c*))

where

assert-sla is a function to check if the KPIs satisfy the service level agreement and *compensate* is a function to make a compensation . Their function signatures are:

assert-sla: *KPI-set* $\times$ *SLA* $\rightarrow$ **B**

compensate: *Remediation* $\rightarrow$ **B**

5 Application of the Accountability Meta-model

The meta-model can be used to model accountability requirements from a CIM and PIM level. As an example here we use the meta-model to model the accountability requirements during the lifecycle of a generic service contract. In the context of BPM and Service-Oriented Architecture (SOA), a process can be implemented as a service, which can be treated as a contract between the service provider and the service consumer. Applying Schedler's definition [8], the service provider is obliged to disclose and justify actions and decisions in relation to the service under the contract. To do that, the service provider must monitor the service, measure the performance and report the service status and service level. The service provider also needs to prepare evidence to justify its position in case of a dispute raised by the service consumer. In this context, misconduct can be interpreted as not meeting the service-level agreement (SLA). Punishment is equivalent to the realization of the remediation clauses such as monetary compensation, service credit or other form of prescribed penalty. Using the meta-model, service provider's accountability can be modeled through a process of disclosure, logging, monitoring, measuring, reporting and remediation in order to manage the risks during the lifecycle of a service contract. Fig. 4 illustrates the accountability process model using BPMN.

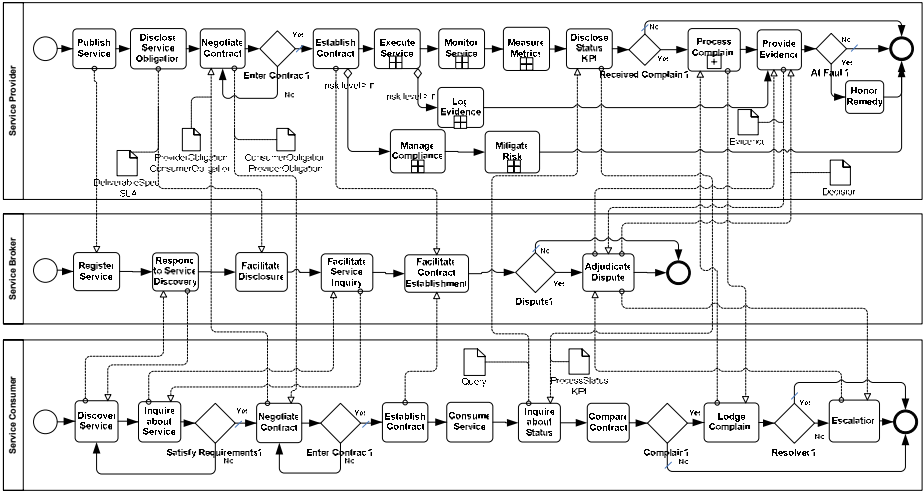


Fig. 4. Accountability Process Model for the Lifecycle of a Service Contract

From a technical perspective, the process model outlines the choreography of the accountability process amongst the service provider, the service broker and the service consumer in a Service-Oriented Architecture setting. In addition, it specifies the orchestration of the high-level accountability tasks such as monitoring and measuring service, which are sub-processes that need to be further modeled on the next level. The sub-process level of modeling is not covered in this paper. Existing approaches can be found in [18], where the authors review the existing approaches and present an MDA approach to model the monitoring and measuring sub-processes. Assuming that the service interfaces for each sub-process and task exist, a PSM accountability model may be generated through a series of model transformations. From a business perspective, the accountability process model also reminds the service provider that providing service always involves extra cost and overhead for accountability related tasks. Thus accountability cost needs to be considered in the overall cost model of the service during the contract negotiation stage. In practice, accountability is normally emphasized when there is a significant risk involved in the service. For example, accountability would be a major issue for a process that deals with patients' records in a medical center's website; whereas it would be less a concern in a process that provides quotes of the day on a free website. Therefore, when applying the accountability meta-model to the lifecycle of a specific service contract, the service provider needs to assess the risks associated with the service, and deploy the accountability tasks in accordance to the risk level. In Fig. 4, some accountability tasks are conditional based on the risk level associated with the service.

6 Conclusion and Future Work

Accountability is increasingly becoming a central concern in business process while more and more corporate scandals and fallouts are revealed in the market. Business

process modeling needs to address the accountability requirements to meet this immediate business concern. The contribution of this paper is to define the first business accountability meta-model based on the OMG's business modeling specifications. The paper also outlines the choreography and orchestration of the accountability process for the lifecycle of a generic service contract based on the accountability meta-model. Positioning itself in OMG's MDA framework, the accountability meta-model enables business analysts to create CIM and PIM process models that address the accountability requirements, which lays out a foundation for the future work on model transformation from PIM to platform specific models.

References

1. OMG: Business Motivation Model, v1.0, Object Management Group (2008), <http://www.omg.org/spec/BMM/1.0/PDF>
2. OMG: Semantics of Business Vocabulary and Business Rules (SBVR), v1.0, Object Management Group (2008), <http://www.omg.org/spec/SBVR/1.0/PDF>
3. OMG: Business Process Definition MetaModel, Volume I: Common Infrastructure, Object Management Group (2008), <http://www.omg.org/spec/BPDM/1.0>
4. OMG: Business Process Modeling Notation, v1.1, Object Management Group (2008), <http://www.omg.org/spec/BPMN/1.1/PDF>
5. 88Solutions, Adaptive and et al: Organization Structure Metamodel (OSM), 2nd Initial Submission, Object Management Group (2006), <http://www.omg.org/docs/bmi/06-11-02.pdf>
6. OMG, Business Rules Representation, Object Management Group (2007), <http://www.omg.org/spec/PRR/1.0/>
7. Oakes, L.S., Young, J.J.: Accountability re-examined: evidence from Hull House. *Accounting, Auditing & Accountability Journal* 21(6), 765–790 (2008)
8. Schedler, A.: Self-Restraining State: Power and Accountability in New Democracies, pp. 13–28. Lynne Reiner Publishers (1999)
9. Gibbins, M., Newton, J.: An empirical exploration of complex accountability in public accounting. *Journal of Accounting Research* 32(2), 165–186 (1994)
10. Zou, J., Pavlovski, C.J.: Towards Accountable Enterprise Mashup Services. In: *IEEE International Conference on e-Business Engineering (ICEBE 2007)*, Hong Kong, pp. 205–212 (2007)
11. Jones, C.B.: *Systematic Software Development Using VDM*, 2nd edn. Prentice-Hall International, Englewood Cliffs (1990)
12. Kailar, R.: Reasoning about Accountability in Protocols for Electronic Commerce. In: *Proceedings of 1995 IEEE Symposium on Security and Privacy*, p. 236. IEEE Computer Society, Los Alamitos (1995)
13. Zhou, J., Gollman, D.: A fair non-repudiation protocol. In: *Proceedings of IEEE Symposium on Security and Privacy*. IEEE Press, Los Alamitos (1996)
14. Robinson, P., Cook, N., Shrivastava, S.: Implementing fair non-repudiable interactions with Web services. In: *Proceedings of Ninth IEEE International EDOC Enterprise Computing Conference*. IEEE Press, Los Alamitos (2005)
15. Tseng, M.M., Chuan, J.S., Ma, Q.H.: Accountability Centered Approach to business process reengineering. In: *Proceedings of the 31st Hawaii International Conference on System Sciences*, vol. 4, pp. 345–354 (1998)

16. Zhang, Y., Lin, K.J., Yu, T.: Accountability in Service-Oriented Architecture: Computing with Reasoning and Reputation. In: Proceedings of IEEE International Conference on e-Business Engineering, pp. 123–131 (2006)
17. List, B., Korherr, B.: A UML 2 Profile for Business Process Modeling. Springer, Heidelberg (2005)
18. Momm, C., Malec, R., Abeck, S.: Towards a Model-driven Development of Monitored Process. In: Internationale Tagung Wirtschaftsinformatik, WI 2007 (2007)
19. OMG, Model Driven Architecture (2008), <http://www.omg.org/mda/>

Appendix A

Table 1. VDM Symbols and Keywords

Keywords / Symbols	Meanings
inv	data invariant
mk	composite object creation function
Δ	definition
$\leftrightarrow^m$	one to one map
set	set of type
B	boolean type
$\tilde{e}$	e 's state prior to entering the function or operation

Extensible and Precise Modeling for Wireless Sensor Networks

Bahar Akbal-Delibas, Pruet Boonma, and Junichi Suzuki

Department of Computer Science
University of Massachusetts Boston
Boston, MA 02125 USA
{abakbal, pruet, jxs}@cs.umb.edu

Abstract. Developing applications for wireless sensor networks (WSN) is a complicated process because of the wide variety of WSN applications and low-level implementation details. Model-Driven Engineering offers an effective solution to WSN application developers by hiding the details of lower layers and raising the level of abstraction. However, balancing between abstraction level and unambiguity is challenging issue. This paper presents *Baobab*, a metamodeling framework for designing WSN applications and generating the corresponding code, to overcome the conflict between abstraction and reusability versus unambiguity. Baobab allows users to define functional and non-functional aspects of a system separately as software models, validate them and generate code automatically.

1 Introduction

Wireless sensor networks (WSNs) are used to detect events and/or collect data in physical observation areas. They have been rapidly increasing in their scale and complexity as their application domains expand, from environment monitoring to precision agriculture, from perishable food transportation to disaster response, as just a few examples. The increase in scale and complexity make WSN application development complicated, time consuming and error prone [1].

The complexity of WSN application development derives from a lack of abstraction. A number of applications are currently implemented in nesC, a dialect of the C language, and deployed on the TinyOS operating system, which provides low-level libraries for basic functionalities such as sensor reading, packet transmission and signal strength sensing. nesC and TinyOS abstract away hardware-level details; however, they do not aid developers to rapidly implement their applications.

Model-driven development (MDD) is intended to offer a solution to this issue by hiding low-level details and raising the level of abstraction. Its high-level modeling and code generation capabilities are expected to improve productivity in WSN application development (e.g., [1, 2, 3]). However, there is a research issue in MDD, particularly in metamodeling, for WSN applications: balancing generalization and specialization in designing a metamodel for WSN applications. When metamodel designers want their metamodels to be as much generic and versatile as possible for various application domains, the metamodels can be over generalized (e.g., [2, 3]).

Over generalized metamodels tend to be ambiguous and type-unsafe. Metamodel users do not understand how to specifically use metamodel elements and often make errors that metamodel designers do not expect. Model-to-code transformers can fail due to ambiguous uses and errors that metamodel users make in their modeling work.

Another extreme in metamodeling is over specialized metamodels (e.g., [1]). Over specialized metamodels can avoid ambiguity and type unsafety; however, it lacks extensibility and versatility. Metamodel users cannot extend metamodel elements to accommodate requirements in their applications and cannot use them in the application domains that metamodel designers do not expect.

Baobab is an MDD framework that addresses this research issue for WSN applications. It provides a generic metamodel (GMM) that is versatile across different application domains. Metamodel users can use it to model both functional and non-functional aspects of their WSN applications. Baobab allows metamodel users to extend GMM for defining their own domain-specific metamodels (DSMMs) and platform-specific metamodels (PSMM). This extensibility is driven with *generics* to attain the type compatibility among GMM, DSMM and PSMM elements as well as the Object Constraint Language (OCL) [4] for avoiding metamodel users to extend GMM in unexpected ways. These two mechanisms allow application models to be type safe and unambiguous. Baobab's model-to-code transformer type-checks and validates a given application model and generates application code in nesC. It can generate most of application code, and the generated code is lightweight enough to operate on resource-limited sensor nodes such as Mica2 nodes.

2 Metamodels and Models for WSN Applications

In Baobab, metamodels are partitioned into different packages. GMM is defined in the *genericMetamodel* package.

2.1 Generic Metamodel Elements

The element Sensor of the GMM represents sensor devices that are used in WSNs. All sensor classes, representing a specific type of sensor, extend from the base class *Sensor*. The most common sensor types that can be used in a variety of applications are defined in the generic metamodel. As the names imply, each sensor detects the specific phenomenon it is prefixed by.

Nodes may send data to each other in a WSN occasionally. This can be done by packing *Data* (either some sensor reading value or a command) in a *Message*, and sending it to other nodes by a *CommunicationUnit*, which consists of a *DataTransmitter* and a *DataReceiver*. A *WirelessLink* represents the communication channel between two *CommunicationUnits*. The sensor readings and the associated information are represented as *SensorData*. Specific classes that extend from *SensorData* will have their own attributes, as well as the shared attributes. For example, *AirTempData* has a *temperature* attribute holding the air temperature reading value. When nodes aggregate multiple *SensorData* instead of transmitting them separately, an *AggregatedData* is created. All types of *Data* can be stored in and retrieved from a *DataStorage* by *DataWriter* and *DataReader*, respectively. *EnergySource* can be used to interrogate the remaining energy level of the node.

Usage of generic types in the GMM increases the extensibility of GMM elements, as well as assuring type-safety. As an example, the *Sensor* in our generic metamodel is expected to create *SensorData*, whereas *AirTemperatureSensor* creates *AirTempData*. We defined the type of *sensorData* reference between *Sensor* and *SensorData* as a generic type that extends from *SensorData* in the generic metamodel. Thus it is feasible to associate *AirTempData* with *AirTemperatureSensor* in the GMM, and associate *BacteriaData* with *BacteriaSensor* later in the fresh-food domain metamodel.

There is a set of tasks that should be performed by a WSN node during each duty cycle. By the end of the duty cycle duration the sensor nodes will go to sleep in order to save energy. A *Timer* and a *DutyCycleManager* in the GMM manage all these series of events. At the beginning of each duty cycle *DutyCycleManager* invokes a chain of tasks to be performed, by calling the *firstTask* of the task chain defined. Each task to be performed is represented as a *Task* in the GMM. Upon completion, each *Task* will call the next *Task* defined. The tasks regarding the functional requirements of the WSN system are encapsulated in *FunctionalTasks*, whereas the tasks regarding the non-functional requirements of the WSN system are encapsulated in *NonFunctionalTasks*.

2.1.1 Functional Requirements

GMM defines several elements to express the most common functional aspects of WSNs. The functional tasks whose execution is bound to the fulfillment of a condition can be modeled by using *ConditionalFunctional* element of the GMM. This task can further be specialized into *RepetitiveTask*, which lets users to model iteration with conditions defined by the comparison of the two attributes: *repetitionNumber*, for holding the desired number of repetitions, and *repeated*, for keeping the number of repetitions completed so far. *DataReceipt* is used to define the receipt of data from another node in the network. In some cases, tasks need to be followed by a waiting period before another task can be called, which can be modeled by using *WaitingTask*. Another common functionality of WSNs, sensing phenomena, can be modeled with *SensingTask*. This element retrieves the newly created *SensorData* from the *Sensor*.

2.1.2 Non-functional Requirements

Non-functional requirements represent the quality goals and constraints of a system. The tolerance rate of service performance, and constraints of a system are likely to change more often than the services (functional requirements) themselves in a system. Therefore, functional and non-functional aspects of a system should be modeled independent from each other. This separation not only enables developers to adapt the existing systems to new non-functional requirements easily, without annulling the whole design and creating a system from scratch, but also enables developers to reuse services in different non-functional contexts for future systems.

The non-functional requirements of a system can be modeled explicitly by means of *NonFunctionalTask* class in our GMM. The specialized non-functional tasks that are defined in the GMM are: *ClusterFormation*, for dividing the network into clusters to simplify tasks such as communication [5] and to save energy by aggregating data within the cluster; *ChangeSleepTime*, for adjusting the sleep time to minimize energy



Fig. 1. Partial Generic Metamodel

consumption or to maximize data collection; *DataAggregation*, for aggregating data to be transmitted for the sake of eliminating redundancy, minimizing the number of transmissions and thus saving energy [6]; *DataTransmission*, for transmitting data with a specific policy based on the non-functional requirements; *ConditionalNonFunctional*, for specifying the activation of a *NonFunctionalTask* that is bound to fulfillment of a condition; and *ChangeCommunicationRange*, for adjusting the physical range of transmission to minimize energy consumption or to maximize data collection.

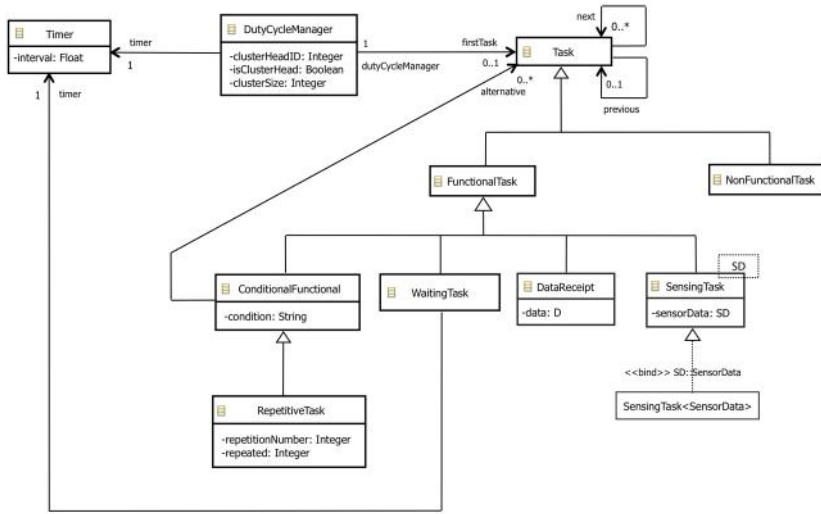


Fig. 2. Functional Aspects of the Generic Metamodel

ClusterFormation has a reference to a *ClusteringAlgorithm*, which can be one of the specialized clustering algorithms [5]. *ChangeSleepTime* and *ChangeCommunicationRange* tasks can be used to adjust the sleep time and communication range, respectively, by a given rate.

DataAggregation task has the attribute domain to specify whether the aggregation will be a *TEMPORAL* aggregation or a *SPATIAL* aggregation. The other attributes of *DataAggregation* are: *hop*, to specify how many hops away neighbors' data will be aggregated (only if *SPATIAL* aggregation domain is selected); *dutyCycleNumber*, to specify the number of duty cycles' collected data to be aggregated (only if *TEMPORAL* aggregation domain is selected); *aggregatingNodes*, the list of *nodeIDs* of the neighboring nodes to aggregate data with (only if *SPATIAL* aggregation domain is selected); and *dataList*, the list of the collected data to be aggregated. The types of aggregation supported in GMM are *Average*, *Minimum*, *Maximum*, *Mean*, *Variance*, *MinimumAndMaximum*, *StandardDeviation*, *Suppression* (eliminating redundant data, e.g. if the temperature readings of all neighboring sensors in a region are same, only one packet containing the single sensor reading will be forwarded to the base station), and *Packaging* (combining similar data into a single message). When using *Packaging*, either of the *timeWindow* or *numberOfData* attributes should be set. Using the attribute *timeWindow* denotes that exactly the same kind of data is packed together (e.g. temperature readings for the last 10 minutes), while using *numberOfData* denotes that different kind of but related data is packed together (e.g. temperature readings and air flow readings).

For *DataTransmission* task, GMM defines four possible communication policies. *Unicast* delivers a message to a single specified node, *Broadcast* delivers a message to all nodes in the network, *Multicast* delivers a message to a group of nodes that have expressed interest in receiving the message and *Anycast* delivers a message to any one out of a group of nodes, typically the one nearest to the source.

metamodel elements. The GMM already has *AirTemperatureSensor* and *AirTemp-Data* so the user does not have to define them again. However, there are no sensors or specific data types defined for airflow and bacteria in the GMM. So, they are added into this new DSMM package. Possible corrective actions to be taken by the base station are: changing the airflow speed and air temperature in the cold chamber. Based on this knowledge, two new functional tasks can be defined.

2.3 Platform Specific Metamodel Elements

The GMM and the DSMM explained in the previous sections are platform-independent, in other words, they do not capture the details of the implementation language, the operating system to be deployed on, or the architecture of the application. This section explains the usage of Platform-Specific Metamodel (PSMM) elements. Separating the DSMM and PSMM results in highly re-usable models. For example, one may want to design a system by using the fresh food DSMM for mica nodes (built upon nesC and TinyOS combination) as the target platform, using a biologically inspired architecture, and then the same domain-specific model can be re-used to design an application to work on SunSPOT (built upon Java and JVM combination), using a database-centric architecture. The metamodel elements and the transformation rules used for the fresh food domain remains the same, but the target platform specifications change.

BiSNET (Biologically-inspired architecture for Sensor NETworks) is a middleware architecture for multi-modal WSNs [9]. The two software components in BiSNET are agents and middleware platforms. Agents sense their local environments, and take actions according to sensed conditions. Upon a significant change in sensor reading an agent (a stationary agent that resides on a platform all the time) emits a pheromone to stimulate replicating itself and its neighboring agents. Each agent replicates only when enough types and concentration of pheromones become available on the local node. A replicated agent (a migratory agent) migrates toward a base station on a hop-by-hop basis to report sensor data.

PSMM elements can be added to GMM just as DSMM elements are added. A new package for each platform should be created under the same directory as GMM, and then the new package can be populated with the necessary PSMM elements extending from GMM elements. Fig. 4 depicts the resulting BiSNET PSMM.

The middleware platform and agent concepts in BiSNET are mapped to PSMM elements *Platform* and *Agent*, respectively. Since there is no entity to model software agents in GMM, *Agent* defined in *bisnet* package does not extend from any element of the GMM. The two types of agents, stationary agents and migratory agents, are mapped to *StationaryAgent* and *MigratoryAgent* in the *bisnet* PSMM, respectively.

2.4 Creating a Model Based on the Metamodel

There are four types of sensor nodes in the application scenario explained above: bacteria node, air temperature node, airflow node and the base station. Since Baobab considers modeling the components and functionalities of each type of node separately, four different models should be created. Fig. 5 depicts the model created for bacteria node.



Fig. 5. Model based on fresh food domain and BiSNET platform

3 Model Validation with OCL

Baobab allows metamodel designers to specify OCL constraints on metamodel elements so that they are extended to DSMMs and PSMMs and instantiated in models in unambiguous manner. OCL constraints can set restrictions on property values and

specify dependencies between property values of an element, or different elements. Then, Baobab validates models with a given set of OCL constraints. Listing 1 shows some of the OCL constraints that are checked against the model depicted in Fig. 5.

4 Model-to-Code Transformation

This section describes how Baobab transforms a model created with GMM, DSMs, and PSMMs into nesC code for TinyOS. Currently, Baobab assumes that all meta-models and models are defined on Eclipse Modeling Framework¹ and uses openArchitectureware² to implement its model-to-code transformer.

Listing 3 is a code snippet that Babab generates from the model depicted in Fig. 5. The code performs five tasks starting with *PheromoneSensingTask*. *PheromoneSensingTask* is performed in the code by calling a BiSNET-specific function, *pheromoneSensing()*, with *pheromoneType* specified in Fig. 5 as a parameter. *DataAggregation* is performed by calling *getAggregatedData()* of the *DataAggregation* interface with relevant parameters specified in Fig. 5. *getAggregatedData()* takes a parameter on *aggregationType*. *AgentMigrationTask* is performed by calling *migrationTask()*, which is another BiSNET-specific function.

As for *AgentReplicationTask*, a conditional expression is generated as a comment in an if-statement. The actual value to be checked if it is greater than two is *aggregatedData[0]* of the previous task (*DataAggregation*). However, the generated code does not keep the previous for a task because it can be any *Task* subtype, but nesC is not an object-oriented language and it does not support polymorphism. Thus, this conditional expression is left as a comment and a programmer should replace it with a real Boolean expression to reflect what is meant in the model.

Listing 1.

```
-- All Data created by a Sensor should be for the same phenomenon.
context Sensor
inv sensorData->forall(a1, a2 | a1 <> a2 and
    a1.phenomenonType = a2.phenomenonType)

-- AirTempSensor can only generate AirTempData.
context AirTempSensor
inv: sensorData->forall(self.ocIsTypeOf(AirTempData))

-- dutyCycleNumber is used only when aggregation domain is TEMPORAL.
-- hop and aggregatingNodes should be only used when domain is SPATIAL.
context DataAggregation
inv: dutyCycleManager <> null implies domain = TEMPORAL
    and hop <> null implies domain = SPATIAL
    and aggregatingNodes <> null implies domain = SPATIAL

-- If aggregation domain is SPATIAL, aggregatingNodes and hop are non-null.
context DataAggregation
inv: domain = SPATIAL implies
    (hop <> null) xor (aggregatingNodes <> null)

-- If aggregation domain is TEMPORAL, dutyCycleNumber cannot be null.
context DataAggregation
inv: domain = TEMPORAL implies dutyCycleNumber <> null
```

¹ www.eclipse.org/modeling/emf

² www.eclipse.org/gmt/oaw

Listing 2.

```
//:PheromoneSensingTask
pheromones = pheromoneSensing(REPLICATION);
//:DataAggregation
aggregatedData = call DataAggregation.getAggregatedData(
                                SPATIAL, 1, 0, AVERAGE);
//:AgentReplicationTask
if(/* previous.aggregatedData[0] > 2 */) {
    int weight[4] = {0, 0, 0, 0};
    call Agent.setWeight(weight);
    agent = replicationTask(
        aggregatedData.sensorData[0].colonyFormingUnitsPg,1);
    // :ChangeSleepTimeTask
    Timer_interval *= 0.5;
    //:AgentMigrationTask
    migrationTask(agent); }
```

5 Preliminary Evaluation

This section discusses preliminary results to evaluate Baobab. Baobab generates 1,279 lines of nesC code from the model depicted in Figure 5. It takes 544 milliseconds to generate the code. After the code is generated, there are 12 lines of code to be manually written by a programmer, which takes approximately 2 minutes. Baobab generates 99.1% of the total code; it can significantly simplify the development of WSN applications. The generated code can be deployed on the Mica2 sensor node as well as the TOSSIM simulator [10]. Table 1 shows memory footprint of the generated code on the two deployment environments. Baobab generates lightweight nesC code that can operate on sensor nodes with severely limited resources. For example, a Mica2 node has 4 KB in RAM and 128 KB in ROM.

Table 1. Memory Footprint of a Generated WSN application

	ROM (bytes)	RAM (bytes)
Mica2	19,496	1,153
PC (TOSSIM)	72,580	179,188

6 Conclusion

This paper proposes an MDD framework, called Baobab, for WSN application development. Baobab provides a metamodel that includes the most common components and behaviors of WSN nodes, in a platform-independent way. Besides, it can be extended easily for new application domains and platforms without impairing the existing elements and rules. This metamodel also enables users to model non-functional aspects of WSN systems as well as their functional aspects. Applications can be constrained further by a set of OCL rules, and the models can be validated against these rules. The model-to-code generator creates runnable code from the input models with a little modification by the programmers.

References

1. Wada, H., Boonma, P., Suzuki, J., Oba, K.: Modeling and Executing Adaptive Sensor Network Applications with the Matilda UML Virtual Machine. In: Proc. of IASTED International Conference on Software Engineering and Applications (2007)
2. Vicente-Chicote, C., Losilla, F., Álvarez, B., Iborra, A., Iborra, P.: Applying MDE to the Development of Flexible and Reusable Wireless Sensor Networks. *International Journal of Cooperative Information Systems* 16(3), 393–412 (2007)
3. Sadilek, D.A.: Prototyping Domain-Specific Languages for Wireless Sensor Networks. In: Proc. of International Workshop on Software Language Engineering (2007)
4. The Object Management Group, Unified Modeling Language (UML) Superstructure and Infrastructure, version 2.1.2 (2007)
5. Dechene, D.J., El Jardali, A., Luccini, M., Sauer, A.: A Survey of Clustering Algorithms for Wireless Sensor Networks. Project Report (2006)
6. Krishnamachari, B., Estrin, D., Wicker, S.: The Impact of Data Aggregation in Wireless Sensor Networks. In: Proc. of Int'l Workshop of Distributed Event Based Systems (2002)
7. United Nations Economic Commission for Europe: UNECE Standard Bovine Meat Carcasses and Cuts. 2007 Edition, United Nations, New York and Geneva (2007)
8. Commission Regulation (EC) No 2073/2005 of 15 November 2005 on microbiological criteria for foodstuffs. *Official Journal of the European Communities* (2005)
9. Boonma, P., Suzuki, J.: BiSNET: A biologically-inspired middleware architecture for self-managing wireless sensor networks. *Computer Networks* 51 (2007)
10. Levis, P., Lee, N., Welsh, M., Culler, D.: TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications. In: ACM Conf. on Embedded Networked Sensor Systems (2003)

Author Index

- Abdullah, Che Zainab 222
 Agt, Henning 328
 Ahmad, Hashim 222
 Ahmad, M.S. 115
 Akbal-Delibas, Bahar 551
 Al-Fedaghi, Sabah S. 438
 Alves de Medeiros, Ana Karla 190
 Aris, Hazleen 355
 Awami, Salah 346

 Bai, Xiaoyan 280
 Bajaj, Simi (Kamini) 404
 Balram, Shyamala 404
 Bauhoff, Gregor 328
 Bell, Tim 240
 Bonačić, Mirjam 340
 Boonma, Pruet 551
 Brumen, Boštjan 340

 Calabretto, Jean-Pierre 346
 Calero, Coral 298
 Cannella, Salvatore 475
 Cartsburg, Mario 328
 Chen, Kuei-Hsien 142
 Chen, Zhen-Yao 65
 Chowdhury, Belal 420
 Ciancimino, Elena 475
 Corporaal, Henk 190

 D'Souza, Clare 420
 Denize, Sara 496
 De Silva, Buddhima 304
 De Vaney, Christopher 539
 Družovec, Marjan 340

 Ermolayev, Vadim 127

 Falcarin, Paolo 121
 Fukazawa, Yoshiaki 526
 Funk, Mathias 190

 Getta, Janusz R. 75
 Ginige, Athula 53, 153, 304
 Ginige, Jeewani Anupama 153
 Gobbo, William 103
 Godlevskiy, Mikhail D. 91
 Griffith, Sharon 229, 252

 Hasegawa, Tetsuo 526
 Hashim, Rugayah 222
 Hawryszkiewicz, I.T. 458
 Henderson-Sellers, Brian 41
 Hodych, Oles 29
 Hölbl, Marko 340
 Hu, Tung-Lai 65
 Huang, Yi 514
 Huebner, Ewa 184
 Hushchyn, Kostiantyn 29

 Jaakkola, Hannu 340
 Jentzsch, Eyck 127

 Kamenieva, Iryna 411
 Kappel, Gerti 315
 Keberle, Natalya 127
 Khalid, Ruzelan 240
 Khan, Khaled M. 393
 Klopčič, Brane 340
 Konda, Ramesh 292
 Kreutzer, Wolfgang 240
 Kumpe, Daniel 328
 Kuo, R.J. 65
 Kusel, Angelika 315
 Kutsche, Ralf 328

 Lan, Yi-Chen 487
 Le, Dung Xuan Thi 367
 Lee, Maria R. 487
 Lee, Sai Peck 202
 Liang, Xufeng (Danny) 53

 Ma, Hui 17
 Maamar, Zakaria 4
 Marmaridis, Ioakim 229, 252
 Marmaridis, Ioakim (Makis) 53
 Martellone, Alessandro M. 508
 Mat Jani, Hajar 202
 Matzke, Wolf-Ekkehard 127
 Milanovic, Nikola 328
 Moraga, M^a Ángeles 298
 Morisio, Maurizio 121

- Müller, Christine 432
 Müller, Willy 469
- Nankani, Ekta 496
 Nash, John 214
 Nemani, Rao R. 292
 Ngu, Anne H.H. 4
 Nikolski, Iouri 29
 Noack, René 268
- Paolino, Luca 508
 Pardede, Eric 367
 Pasichnyk, Volodymyr 29
 Pataricza, András 1
- Qumer, Asif 41
- Raibulet, Claudia 103
 Reiter, Thomas 315
 Repka, Victoriya 411
 Retschitzegger, Werner 315
 Rosemann, Michael 3
 Rozinat, Anne 190
 Ruan, Chun 184
- Sabar, N.R. 115
 Sakr, Sherif 379
 Samaranayake, Premaratne 165
 Sampath, Partha 178
 Schewe, Klaus-Dieter 17
 Schmid, Beat 469
 Schroth, Christoph 469
 Schwinger, Wieland 315
 Sebillio, Monica 508
 Shatovska, Tetyana 411
 Shcherbyna, Yuri 29
 Shekhovtsov, Vladimir A. 91
 Sheng, Quan Z. 4, 121
 Simoff, Simeon 496
- Sohnius, Richard 127
 Somhom, Samerkae 262
 Su, Chwen-Tzeng 142
 Sugunsil, Prompong 262
 Sundaram, David 280
 Suzuki, Junichi 551
 Swatman, Paula M.C. 346
- Tacconi, David 508
 Tan, Calvin 393
 Thalheim, Bernhard 17
 Tomilko, Yuriy 91
 Tortora, Genoveffa 508
 Turkey, Ayad.M. 115
- Ubezio, Luigi 103
- van der Aalst, Wil 190
 van der Putten, Piet 190
 Vitiello, Giuliana 508
 Vossough, Ehsan 75
- Wang, Yan 539
 Welzer, Tatjana 340
 White, David 280
 Williams, Robert 214
 Wimmer, Manuel 315
 Wirsing, Martin 178
 Wong, Jui-Tsung 142
- Xiao, Jian 514
 Xiao, Lan 514
- Young, Louise 496
 Yu, Jian 4, 121
 Yusoff, M.Z.M. 115
- Zheng, Li 514
 Zou, Joe 539