COMPUTING edge

- Security & Privacy
- Algorithms
- Software Testing
- History

APRIL 2025

www.computer.org

IEEE COMPUTER SOCIETY

IEEE

COMPUTING edge

IEEE COMPUTER SOCIETY

IEEE

IEEE COMPUTER SOCIETY computer.org

Printed with inks containing soy and/or vegetable oils

STAFF

Circulation: *ComputingEdge* (ISSN 2469-7087) is published monthly by the IEEE Computer Society. IEEE Headquarters, Three Park Avenue, 17th Floor, New York, NY 10016-5997; IEEE Computer Society Publications Office, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720; voice +1 714 821 8380; fax +1 714 821 4010; IEEE Computer Society Headquarters, 2001 L Street NW, Suite 700, Washington, DC 20036.

Postmaster: Send address changes to *ComputingEdge*-IEEE Membership Processing Dept., 445 Hoes Lane, Piscataway, NJ 08855. Periodicals Postage Paid at New York, New York, and at additional mailing offices. Printed in USA.

2025 IEEE Computer Society Magazine Editors in Chief

Computer
Jeff Voas, *NIST*

Computing in Science & Engineering
Jeffrey Carver, *University of Alabama*

IEEE Annals of the History of Computing
Troy Astarte, *Swansea University*

IEEE Computer Graphics and Applications
Pak Chung Wong, *Trovares and Bill & Melinda Gates Foundation (Interim EIC)*

IEEE Intelligent Systems
Bo An, *Nanyang Technological University*

IEEE Internet Computing
Weisong Shi, *University of Delaware*

IEEE Micro
Hsien-Hsin Sean Lee, *Intel Corporation*

IEEE MultiMedia
Balakrishnan Prabhakaran, *University of Texas at Dallas*

IEEE Pervasive Computing
Fahim Kawsar, *Nokia Bell Labs and University of Glasgow*

IEEE Security & Privacy
Sean Peisert, *Lawrence Berkeley National Laboratory and University of California, Davis*

IEEE Software
Sigrid Eldh, *Ericsson, Mälardalen University, Sweden; Carleton University, Canada*

IT Professional
Charalampos Z. Patrikakis, *University of West Attica*

APRIL 2025 · VOLUME 11 · NUMBER 4

COMPUTING edge

Security & Privacy

Algorithms

Software Testing

History

Departments

Magazine Roundup

The IEEE Computer Society's lineup of 12 peer-reviewed technical magazines covers cutting-edge topics ranging from software design and computer graphics to Internet computing and security, from scientific applications and machine intelligence to visualization and microchip design. Here are highlights from recent issues.

Computer

Physiological Data: Challenges for Privacy and Ethics

In this article, featured in the January 2025 issue of *Computer*, the authors discuss the potential for wearable devices to be appropriated in ways that extend far beyond their original purpose. They identify how the current technology can be misused, explore how pairing physiological data with nonphysiological data can expand the predictive capacity, and discuss the implications.

Computing in Science & Engineering

Predicting Links in Knowledge Graphs with the Canonical Correlation Analysis and Fusing Tensor Model

Relation prediction in knowledge graphs is critical for uncovering missing links between entities. Previous models mostly focused on learning the distance of entities and relation within each triplet. However, they relied heavily on linear metric learning-based methods to evaluate the connections between them, which ignore high-level complex interactions. To address these problems, the authors of this October–December 2024 *Computing in Science & Engineering* article introduce a Canonical correlation Analysis and Fusing Tensor model (CAFT) for relation prediction.

IEEE Annals of the History of Computing

Developing and Using CAD/CAM/CAE Systems in Boeing

Application programs to improve the quality and performance of its aerospace products are a critical part of Boeing's computing environment. This article, featured in the October–December 2024 issue of *IEEE Annals of the History of Computing*, focuses on how the company developed its own modeling, manufacturing, and engineering programs and built custom software to address shortcomings in commercial, off-the-shelf systems. It also details Boeing's attempt to produce its own computer-aided design system.

IEEE Computer Graphics and Applications

Enhancing Virtual Reality Training Through Artificial Intelligence: A Case Study

In this November/December 2024 *IEEE Computer Graphics and Applications* article, the authors propose an architecture that aims to facilitate the integration of artificial intelligence (AI) assistance into virtual reality (VR) training environments to improve user engagement and reduce authoring effort. The proposed architecture was tested in a study that compared a virtual training session with and without a digital assistant powered by AI.

IEEE Intelligent Systems

Regulated Federated Learning Against the Effects of Heterogeneity and Client Attacks

Federated learning (FL) can complete a learning task without compromising user privacy. However, the FL mechanism, where clients train models using personal data locally and exchange model updates instead of raw data, gives rise to new challenges. The problems

caused by data heterogeneity and malicious client behaviors are universal in practical applications. The authors of this November/December 2024 *IEEE Intelligent Systems* article propose a regulated FL (ReguIFL) that introduces a generator and uses weighted aggregations to regulate client model training and complete federated aggregation.

IEEE Internet Computing

AI Design: A Responsible Artificial Intelligence Framework for Prefilling Impact Assessment Reports

Impact assessment reports for high-risk artificial intelligence (AI) systems will be legally required but challenging to complete, especially for smaller companies. That is because the current process is complex, costly, and relies on guidebooks with limited assistance. The authors of this article from the September/October 2024 issue of *IEEE Internet Computing* propose AI Design, a semiautomatic framework for prefilling these reports.

IEEE micro

AMD XDNA NPU in Ryzen AI Processors

The authors of this article featured in the November/December 2024 issue of *IEEE Micro* discuss the AMD Ryzen 7040 series, the first x86 processor with an integrated neural processing unit (NPU). The artificial intelligence (AI)-optimized capabilities of the Ryzen 7040 NPU enable new AI experiences that are not possible without XDNA, making it a fundamental component in today's Ryzen-AI-powered devices and setting the foundation for an exciting roadmap toward future AI capabilities in mobile PCs.

IEEE MultiMedia

Multimodal Agents: From Vision to Reality

This October–December 2024 *IEEE MultiMedia* article explores the evolution of multimodal agents, highlighting their ability to transcend the limitations of single-modality systems and deliver results based on a comprehensive, context-aware understanding of their environment.

IEEE pervasive COMPUTING
MOBILE SYSTEMS | UBIQUITOUS COMPUTING | INTERNET OF THINGS

The Future of Consumer Edge-AI Computing

In the last decade, deep learning has rapidly infiltrated the consumer end, due to hardware acceleration across devices. The authors of the July–September 2024 issue of *IEEE Pervasive Computing* introduce a novel paradigm centered around EdgeAI-Hub devices, designed to reorganize and optimize compute resources and data access at the consumer edge.

IEEE SECURITY & PRIVACY

Android Permissions: Evolution, Attacks, and Best Practices

In this article, featured in the November/December 2024 issue of *IEEE Security & Privacy*, the author studies the evolution of Android permissions. She describes the rationale behind key changes in Android's permission model and discloses two permission-related security vulnerabilities she discovered. Finally, she provides developers actionable insights to proactively address permission-related security and privacy risks during development.

IEEE Software

Toward an Open Source MLOps Architecture

The authors of this article from the January/February 2025 issue of *IEEE Software* present a Kubernetes-based, open source MLOps framework to streamline the

lifecycle management of machine learning models in production environments. They compare state-of-the-art MLOps tools and frameworks, demonstrating that their features meet the same features as proprietary options, such as Amazon SageMaker.

IT Professional

MetaDigiHuman: Haptic Interfaces for Digital Humans in the Metaverse

As technology continues to advance, the demand for sophisticated and immersive interfaces to interact with the metaverse has become increasingly crucial. This November/December 2024 *IT Professional* article introduces the concept of MetaDigiHuman, a groundbreaking framework that combines blended digital humans and haptic interfaces. By harnessing cutting-edge technologies, MetaDigiHuman enables seamless and immersive interaction within the metaverse.

ADVERTISER INFORMATION

Advertising Coordinator

Debbie Sims
Email: dsims@computer.org
Phone: +1 714-816-2138 | Fax: +1 714-821-4010

Advertising Sales Contacts

Mid-Atlantic US, Northeast, Europe, the Middle East and Africa:
Dawn Scoda
Email: dscoda@computer.org
Phone: +1 732-772-0160
Cell: +1 732-685-6068 | Fax: +1 732-772-0164

Southwest US, California:
Mike Hughes
Email: mikehughes@computer.org
Cell: +1 805-208-5882

Central US, Northwest US, Southeast US, Asia/Pacific:
Eric Kincaid
Email: e.kincaid@computer.org
Phone: +1 214-553-8513 | Fax: +1 888-886-8599
Cell: +1 214-673-3742

Midwest US:
Dave Jones
Email: djones@computer.org
Phone: +1 708-442-5633 | Fax: +1 888-886-8599
Cell: +1 708-624-9901

Jobs Board (West Coast and Asia), Classified Line Ads

Heather Buonadies
Email: hbuonadies@computer.org
Phone: +1 623-233-6575

Jobs Board (East Coast and Europe), SE Radio Podcast

Marie Thompson
Email: marie.thompson@computer.org
Phone: +1 714-813-5094

Editor's Note

Humanmade Securities and Vulnerabilities

As technology becomes smarter and more capable of independent function, it is important to remember where all technology comes from—humans. Humans create cybersecurity programs and the algorithms behind artificial intelligence (AI). Humans generate the data that AI uses to communicate and make decisions. This issue of *ComputingEdge* reminds us of the importance of considering human influence in technological design, particularly for security and privacy. This includes evaluating human behavior when developing security programs, acknowledging the impacts of human bias in algorithm and AI design, and humanizing data by remembering its origins. The articles also examine privacy engineering, software and security testing, and the legacy of inventor Monte Sala.

To implement security practices effectively, companies must understand how to design better cybersecurity and privacy programs. In *IEEE Security & Privacy* article "Leveraging the Human Factors Discipline for Better Cybersecurity Outcomes: A Roundtable Discussion," three experts explain the importance and benefits of considering human factors in cybersecurity design. The authors of "Privacy Engineering From Principles to Practice: A Roadmap," from *IEEE Security & Privacy*, shed light on underrepresented aspects of privacy engineering with the goal of bringing wider understanding and improved use to the discipline.

As algorithms become more important in the world, it is essential to improve their accuracy and objectivity to enable fairer and more efficient algorithmic decision-making. *Computer* article, "From Concept to Reality: Leveraging Correctness-by-Construction for Better Algorithm Design," introduces correctness-by-construction (CbC) development, which facilitates more correct and efficient algorithm design. The article, "Data's Impact on Algorithmic Bias," from *Computer*, reveals how artificial intelligence can generate unfair results and inequalities due to algorithmic bias.

GenerativeAI (GenAI) and machine learning (ML) can enhance software and security testing if used effectively and in a way that does not create more privacy concerns. "Generative Artificial Intelligence and the Future of Software Testing," from *Computer*, discusses the applications of GenAI to software testing and the trust, privacy, and ethical risks involved. The authors of *IEEE Security & Privacy* article, "Why Is Static Application Security Testing Hard to Learn?" demonstrate their approach of combining static analysis and ML techniques to detect security vulnerabilities.

Reflecting on history can help illuminate the origins and meaning of modern data practices, from encryption to use. In "Monte Sala's Cryptographic Achievements," from *IEEE Annals of the History of Computing*, the author chronicles the achievements of Monte Sala, famous for inventing devices for encrypting data. *IEEE Annals of the History of Computing* article "Dissecting Data: History of Data as History of the Body" connects data to the body, reflecting on the corporeality of data as well as data's engagement with and origins from the body.

EDITORS: Yasemin Acar, yasemin.acar@uni-paderborn.de
Jens Grossklags, jens.grossklags@in.tum.de
Julie Haney, julie.haney@nist.gov
Heather Richter Lipford, richter@uncc.edu

This article originally appeared in IEEE SECURITY & PRIVACY vol. 22, no. 6, 2024

DEPARTMENT: THE HUMAN FACTOR

Leveraging the Human Factors Discipline for Better Cybersecurity Outcomes: A Roundtable Discussion

Margaret Cunningham, *Wethos AI*
Calvin Nobles, *University of Maryland Global Campus*
Nikki Robinson, *Capitol Technology University*
Julie Haney, *National Institute of Standards and Technology*

Three human factors experts get to the bottom of what the human factors discipline actually is, how the cybersecurity community and organizations can benefit from it, and how to create a pipeline of professionals with human factors and cybersecurity expertise.

To shed light on the field of human factors and its important role within cybersecurity, Julie Haney, who leads the National Institute of Standards and Technology's Human-Centered Cybersecurity program, facilitated a virtual roundtable with three human factors experts. These experts—whose experiences span academia, government, and industry—directly apply their knowledge of human factors to improve organizational cybersecurity practices and outcomes and train the future cybersecurity workforce.

The roundtable conversation was transcribed, condensed, and edited by and with the approval of the participants.

Dr. Julie Haney: *Can you describe the discipline of human factors and its application within the cybersecurity context?*

Dr. Calvin Nobles: When it comes to human factors, especially in cybersecurity, the definition is not standardized. I've broken it up into two definitions. There's a working definition in which *human factors* refers to any type of behavior that's adverse to a cybersecurity policy or program, increases risk, or makes the program more vulnerable. That is what most people seem to be calling *human factors*. But those of us who practice human factors as a science believe that human factors is about designing a system that accounts for human weaknesses and limitations and improves and optimizes human behavior and performance to the design of that system based on human weaknesses. This second definition can be applied to most domains. But for some reason, in cyber, we have many definitions out there. Because of that, we have a hard time getting people to understand that human factors is based on really three things: it's a science, it's a discipline, and it's a profession.

Dr. Margaret Cunningham: In cybersecurity, people refer to a singular *human factor*. What they are really talking about is the ways that you can count mistakes or the ways that you can say this failure was human. So, the human factor is often seen as human failure. However, those people have missed an opportunity to understand systemic design factors that have impacted human performance. The discipline of human factors is difficult to explain. I think of it as the design of everything from endpoint human–computer interaction to the systems that contribute to that and the environmental factors that impact people.

Digital Object Identifier 10.1109/MSEC.2024.3478348
Date of current version: 11 November 2024

ROUNDTABLE PARTICIPANTS

Margaret Cunningham is the cofounder and chief scientist of Wethos AI as well as an applied experimental psychologist specializing in human performance metrics and behavioral analytics. She holds multiple patents in behavioral risk scoring and predictive analytics and has expertise across behavioral science, performance metric development, product R&D, data security, and human-centric design. Her prior experience includes behavioral engineering at Robinhood, global analytics product management at Forcepoint, and consulting as a human systems integration specialist for the Department of Homeland Security.

Calvin Nobles is the portfolio vice president and dean of the School of Cybersecurity and Information Technology at the University of Maryland Global Campus. He completed fellowships at the Harvard University Belfer Center and the New America Think Tank. In January 2025, he will assume the role of chair of the Human Factors and Ergonomics Society Cyber Technical Group. His unique experiences also include being a commercial-rated pilot and an author. Calvin earned Ph.D.s in human factors and offensive cyberengineering.

Nikki Robinson is a senior technical staff member and lead security architect at IBM. She is also an adjunct professor and doctoral chair at Capitol Technology University. She holds a D.Sc. in cybersecurity as well as a Ph.D. in human factors. She has authored two books, *Effective Vulnerability Management* and *Mind the Tech Gap*. Her research primarily focuses on vulnerability chaining, human factors security engineering, incident response, and threat intelligence. Nikki has more than 15 years of experience in both IT and cybersecurity operations.

Julie Haney is a computer scientist at the National Institute of Standards and Technology, where she leads the Human-Centered Cybersecurity program. Her research interests include the work practices of cybersecurity professionals and the usability and adoption of cybersecurity solutions. Previously, she worked for more than 20 years as a cybersecurity practitioner and technical director at the U.S. Department of Defense. Julie holds a Ph.D. in human-centered computing.

Dr. Nikki Robinson: Traditionally, we think of three different disciplines that come under human factors: engineering, design, and psychology. In the cybersecurity context, one of the things I hear a lot associated with human factors is security awareness training. And while that is one small component of what *human factors* means in security, it's more about understanding the whole human and how we help our users be really effective and use technology and tools well. And then, for security practitioners, how do we build better tooling to help them do their jobs more effectively? How do we automate specific things? How do we understand how practitioners use the systems so that we can continue to improve the type of technology that we're giving them?

Cunningham: In cybersecurity, we often neglect how to build systems that amplify human strengths and allow people to do their best.

Haney: *What are some of the biggest human factors challenges in organizational cybersecurity programs today? How can the discipline of human factors help address these challenges?*

Cunningham: The invisible nature of cybersecurity work, the small team sizes, and the inability to track decisions or make choices or collaboration visible are all big issues to me. We are seeing some growth in that area. Things that I've been excited to see are some of

the workflow, communication, and automation products that can help people who are in distributed teams have a shared mental model of what's going on and some visibility into the actions or decisions that people have made. But it's still a challenge.

Robinson: One of the things that we don't see a lot is measurement. For example, we don't measure perception as a risk. My perception of a risk and someone else's perception of a risk are going to be a little bit different, depending on our experience. I always use an example of a security person working with an IT operations individual responsible for vulnerability management. If they don't have a good working relationship because of frustration in the past, when a security patch affected operations, and the IT individual got in trouble for that, it could actually impact the time to remediation of subsequent vulnerabilities because of their potentially fractured relationship. But we don't really have a good way to measure or identify fractured relationships or perception as a risk control. I think that's one of the bigger challenges. It's not just about the technical control. It could be about the people that are trying to implement that technical control.

In my example, a possible solution is even as simple as awareness—understanding that, sometimes, the time to remediation is not about the difficulty to actually implement the fix, but it could be because the teams are not operating well together and that there's actually some sort of discord that's impacting that ability to remediate. If you could find someone to act as a liaison, someone who understands IT, development, management, security—they don't have to be super technical, but someone who is an effective communicator—that skill set could completely turn the team around. It could completely change the team dynamic. They could come in as a mediator to figure out the challenges each side is having and how to find some common ground. Something as simple as that could really change the risk posture of an organization.

Nobles: The team dynamic piece is huge. I also think there's a lot to learn from other domains about how they approach things. For instance, surgeons used to accidentally leave sponges in patients, but now they have procedures in place. Somebody on that team is responsible for counting sponges and the instruments to make sure they're not inside of the patient. But I think cybersecurity people don't think about human factors. They look for other fixes. I think this is where a human factors expert can really help a team identify friction points. Most people think human factors is something that you can open up a drawer, pull it out and apply, and it works. They don't understand that you have to research the problem to drive solutions.

Cunningham: Taking the time to do that research in your working context is critical. I started my career in human factors in health care. One of the things that we did was we went into the hospitals, and we realized that we could take some of the constructs from fields like aviation that were much more established at that time. But we had to do the work. We had to go in and really understand the tasks even though we could use research methods from aviation, like task analysis and tracking and measuring human performance indicators. And so, embedding someone with the human factors research skill set within your security team is important. I'm so thankful to have champions for human factors science. It's often an engineer who just loves it, but it doesn't necessarily replace someone who knows the research methods and the issues with experimental design and applied settings.

Nobles: I recently had the pleasure of talking with some information security officers. To what Nikki said earlier, when you say *human factors* to these security officers, they run toward security awareness. There's nothing wrong with security awareness. But I believe security awareness is a byproduct of having or not having strong human factors practices. I asked one security officer, "What month do all your people do their cybersecurity training?" He said, "We do it in March." I asked, "When are you going to do it again?" He said, "The following March." I said, "You don't think the bad guys know that? If you have your training in March, research tells us that after about four to eight weeks, if you don't have reinforced training, it tapers off. So, the bad guys have 10 months to play on your employees' weaknesses and human limitations because you're not reinforcing the training enough. They will change their tactics, techniques, and procedures based on your training cycle to deceive your people." I recommended quarterly training or breaking it up monthly to where employees only do

15 min of training at a time. When you see new threats and vulnerabilities apply to your organization, then train on those so that employees don't have to wait until the next training cycle comes around. But most people don't think about it that way. Security awareness is, at best, checking a box, let's be honest.

Haney: *How do we build recognition within organizations and the cybersecurity workforce that human factors matter?*

Cunningham: It's the return on investment. You've got to have a way to communicate that. It's hard to do. Our security teams are scrounging for budget or to buy new technology and increase headcount. And when you say *behavioral engineering*, *human factors engineering*, *human factors* anything, people say, "Yeah, but I need another person to be on my detection and response team. I need another person in application security. And I don't know what I will get from somebody who does human factors engineering." They think human factors are covered in training.

If we think about how to get people to commit to something, we have to deal with how fast something can be done. You've got to have a timeline for it. You have to have a scope for that work. And you have to understand how much it will cost. For practitioners and scientists in this space, helping them define a scope that's narrow enough and can have an improvement that's measurable fast enough is the thing. You must also step back and ask, What is this organization ready for? What is the area of highest impact? How can I communicate what I'm doing in that space? How can I translate it to a metric or an outcome that is already there? Getting that business savviness is the trick. Academics have a great opportunity to take their translational skill set into a business: understanding the problem, making it concrete, taking action or doing an intervention, and then measuring the outcomes. It might look like a product requirements document. It might look like a design document. If you can "businessify" human factors and anchor it on monetary value, you have a chance.

Nobles: One of the biggest things that I notice is that words matter. I think we have to change the terminology. If I come into a room and start talking about human factors, people don't really warm up to that in most cases. They are slow to really want to engage because they feel like it is a very touchy-feely subject. But if I come in a room and say *human factors engineering*, that's different because I'm talking to technical people, and I'm talking in a language that they understand. They look at things from an engineering perspective, like cybersecurity engineering, network engineering, computer engineering, software engineering. I ask them, "What engineering discipline is missing?" And they don't know. I then ask, "If you have a human problem, who are you calling?" And most people say, "Human resources" (HR). Well, HR can't help you if you're talking about reducing the friction around the human element. You need to be looking at a human factors engineer. A human factors engineer is the person that I think is missing from the cybersecurity team right now that can really help chief information security officers (CISOs) fix issues and frustrations around human performance, like distractions, stress, fatigue, and burnout. CISOs have never been trained as technical leaders to look for these things. This is why a human factors professional partnering with your senior security architect is the duo that's going to help bring human factors to life in cybersecurity.

I think we need to get organizations to understand that there's a psychological aspect to cybersecurity. It's okay to say we're going to bring in some psychologists or cognitive behavior analysts to help us understand how people are performing in a very technical, complex environment. I don't think we talk about this subject quite enough, and we don't talk to the right people about it. We have to start talking to the decision makers and helping them understand that they have a knowledge gap. And because they have a knowledge gap, it's going to cascade down through the hierarchy of the organization. We need to start shoring up that knowledge gap.

Haney: *How can the cybersecurity community integrate more human factors knowledge?*

Nobles: People ask me all the time about how to find a human factors engineer. There is not a human factors expert sitting on a bench waiting for the coach to put them in. So, we have to start thinking about how to create a pipeline of people that have expertise in cybersecurity and human factors. There aren't many of us who do both. So, this is a real discussion that needs to happen.

Part of the struggle is the education piece. Every school with a cybersecurity program should be teaching a human factors course because it is a real-world issue that organizations are struggling to deal with today. Or we might need to talk about how we can take cybersecurity professionals and send them to school to get a master's degree in human factors. Many people have taken some type of human–computer interaction (HCI) course, and they will say that's human factors. But what I tell people is to look at human factors as the size of a basketball. Researchers have gone in and cut out a small portion of that basketball to create HCI. HCI is the size of a softball. HCI is very important, and it's gotten us a long way, but some of the fundamentals of human factors are still missing.

Robinson: We see a lot more universities offering a human factors security engineering course. It's something to give security engineers—people who are going to be our future defenders, our new security analysts and engineers and architects—a bit of what *human factors* means. That may even spark interest in them to say, "Oh, this research or book is kind of interesting. Maybe I'll integrate this into what I do when I am a practitioner." I think giving them the information upfront before they join the cyber workforce can at least help them think about things in a different way, not just always see something as a technology problem. We've gotten really good about educating people on technology, giving them the hands-on training, giving them the tools that they need to get going when they join the workforce. But I think incorporating that human factors piece, even just one course in that program, could make a huge difference.

One of the other things that we can do is encourage collaboration between the academic community and the private sector. I've definitely seen more of that. However, as someone who is a practitioner as well as an academic person, sometimes there's this stigma that comes with having a Ph.D. in human factors or Ph.D. at all when it comes to the technical community. But when I get to work with both practitioners and academics, I get to see challenges from both sides. I get to see some solutions from the academic community, but I also get to see the problem sets from the technical community, and I can blend both. I think there's a lot to be learned from bringing in both perspectives. Human factors security research goes back almost 20 years, but we just haven't seen it integrated into what we do as practitioners. So, there's more collaboration we can do there to really learn from each other.

Haney: *How can organizations get started toward incorporating human factors into their cybersecurity programs?*

Cunningham: The hard thing is that every company is a living organism. They've got many different tools, many different choices, different people. And so, that does pose some challenges. I would first attend technical onboarding for new engineers, both the builders and the security engineers. And I would listen to all of the different things that they're taught about the system that they're going to work on. And I would listen very specifically for a few key terms that serve as a flashlight on human factors issues: "Be careful," "Pay attention," "You can ignore that," "This is where we have mistakes." If you can find those in your teams, you have insight into something broken. I would also look for internal corporate engineering or IT tickets. I would read all of them. I would say, "Okay, well, this is a thing that breaks in your company all the time. This problem has consistently come up when a new person joins; this is their roadblock." And all of those little things are typically actionable. It's a physical system that people are struggling with.

Nobles: The help desk is my favorite place.

Robinson: I totally agree. In addition, my recommendation for any organization that wants to explore human factors is to do a consultation with a human factors security expert and see what they have to say. Even something as simple as that can give you a good idea of what to start with. And once organizations start to see the benefits of looking at a problem in a different way, that may encourage them to hire a human factors expert, build their own program, and build a human factors expert into the security team. Start small and take bite-sized chunks.

Robinson: My last point would just be that human factors security research is out there. I encourage anyone that's interested in the subject to look up human factors security engineering. There's already some good information to get you going.[1,2,3,4,5]

Cunningham: I think that you can sometimes find secret human factors people. They're doing human factors work, like implementing an architecture so that you eliminate a human performance issue. So, look for people who are already doing that work and start naming them. Find your partners.

Nobles: We need to continue to reinforce why the field of human factors is important for cybersecurity. If we stop talking about it, people are going to assume it's no longer an issue or it has been resolved. I want to be an advocate until the role of a human factors engineer becomes normalized, like how a software engineer is a thing. No one 15 to 20 years ago was talking about software engineering the way we talk about software engineering today. So, you've seen the evolution of software engineering over our lifetimes. And I just want to make sure that we continue to have a similar conversation about human factors in cybersecurity.

REFERENCES

1. C. Nobles, "Stress, burnout, and security fatigue in cybersecurity: A human factors problem," *HOLISTICA–J. Bus. Public Admin.*, vol. 13, no. 1, pp. 49–72, Jun. 2022, doi: 10.2478/hjbpa-2022-0003.
2. C. Nobles, "Human factors in cybersecurity: Academia's missed opportunity," in *Proc. Nineteenth Midwest Assoc. Inf. Syst. Conf.*, Saint Paul, Minnesota, 2023, pp. 8.
3. C. Nobles and D. Burrell, "Exploring the variability of human factors definitions in cybersecurity literature," in *Proc. Nineteenth Midwest Assoc. Inf. Syst. Conf.*, Peoria, Illinois, 2024, p. 28.
4. C. Nobles, N. Robinson, and M. Cunningham, "Straight from the human factors professionals' mouth: The need to teach human factors in cybersecurity," in *Proc. 23rd Annu. Conf. Inf. Technol. Educ.*, 2022, pp. 157–158, doi: 10.1145/3537674.3555782.
5. N. Robinson, "Human factors security engineering: The future of cybersecurity teams," *EDP Audit Control Secur. Newslett.*, vol. 67, no. 5, pp. 1–17, Jun. 2023, doi: 10.1080/07366981.2023.2211429.

MARGARET CUNNINGHAM is the cofounder and chief scientist of Wethos AI, Austin, TX 78731 USA. Her research interests include human factors engineering, behaviorial analytics, and cybersecurity. Cunningham received a Ph.D. in applied experimental psychology from The Catholic University of America. Contact her at margaret@thehumanconsultant.com.

CALVIN NOBLES is the portfolio vice president and dean of the School of Cybersecurity and Information Technology at the University of Maryland Global Campus, East Adelphi, MD 20783 USA. His research interests include the human element in cybersecurity, human performance issues in cybersecurity, and the weaponization of artificial intelligence in cybersecurity. Nobles received a Ph.D. in human factors and a Ph.D. in offensive cyberengineering from Capitol Technology University. He is a member of the Chartered Institute of Ergonomics and Human Factors, Human Factors and Ergonomics Society, and ISACA. Contact him at calvin.nobles@umgc.edu.

NIKKI ROBINSON is a senior technical staff member and lead security architect at IBM as well as an adjunct professor and doctoral chair at Capitol Technology University, Laurel, MD 20708 USA. Robinson received a D.Sc. in cybersecurity and a Ph.D. in human factors from Capitol Technology University. Her research interests include vulnerability chaining, human factors security engineering, incident response, and threat intelligence. Contact her at nerobinson@captechu.edu.

JULIE HANEY is a computer scientist at the National Institute of Standards and Technology, Gaithersburg, MD 20899 USA, where she leads the Human-Centered Cybersecurity program. Haney received a Ph.D. in human-centered computing from the University of Maryland, Baltimore County. Her research interests include the work practices of cybersecurity professionals, the usability and adoption of security solutions, and people's perceptions of security and privacy. Contact her at julie.haney@nist.gov.

DISCLAIMER

Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of their employers, the National Institute of Standards and Technology, or the U.S. Government.

EDITORS: Roberto Baldoni, baldoni@diag.uniroma1.it
Khaled El Emam, kelemam@cheo.on.ca

This article originally appeared in
IEEE SECURITY & PRIVACY
vol. 22, no. 2, 2024

DEPARTMENT:
SECURITY AND PRIVACY GOVERNANCE

Privacy Engineering From Principles to Practice: A Roadmap

Frank Pallas, *Technical University Berlin*
Katharina Koerner, *Daiki*
Isabel Barberá, *Rhite*
Jaap-Henk Hoepman, *Radboud University*
Meiko Jensen, *Karlstad University*
Nandita Rao Narla, *DoorDash*
Nikita Samarin, *University of California, Berkeley*
Max-R. Ulbricht, *Berlin Commissioner for Data Protection and Freedom of Information*
Isabel Wagner, *University of Basel*
Kim Wuyts, *PriceWaterhouseCoopers Belgium*
Christian Zimmermann, *Robert Bosch GmbH*

Privacy engineering is gaining momentum in industry and academia alike. So far, manifold low-level primitives and higher-level methods and strategies have successfully been established. Still, fostering adoption in real-world information systems calls for additional aspects to be consciously considered in research and practice.

With organizations facing increasingly stringent data protection regulations and digital trust being at the heart of growing user expectations, privacy engineering is gaining traction as a distinct discipline in business and academia alike. Large enterprises are establishing dedicated privacy engineering departments and more and more scientific venues are adopting privacy engineering as one of their central themes, confirming Lea Kissner's and Lorrie Cranor's 2021 designation of privacy engineers as the "superheroes" of the privacy profession.[1]

Privacy engineering leverages concepts from disciplines as diverse as information security, jurisprudence, economics, and psychology to facilitate the development of systems that are privacy-friendly by design. It explicitly takes a comprehensive view of such systems and services, as well as their development and socio-technical surroundings. This helps bridge the gap between practical implementations and traditional privacy and security research. It allows companies to better and more reliably comply with increasing enforcement of regulations, such as the European General Data Protection Regulation (GDPR) or the California Privacy Rights Act (CPRA). Privacy engineering also helps companies increase trust in their data-handling practices on the side of their customers, employees, and business partners, as well as to demonstrate accountability to data protection authorities.

While a vibrant community of academic researchers and corporate privacy engineers have been progressing the field significantly during the last years, uptake in the industry at large is still relatively low. Even though numerous design methods and frameworks have been established—from privacy threat

Digital Object Identifier 10.1109/MSEC.2024.3363829
Date of current version: 15 March 2024

 Published by the IEEE Computer Society

modeling frameworks, such as LINDDUN[2] to generic privacy design strategies[3]—privacy is, broadly speaking and with the exception proving the rule, still no first-class member of modern, real-world information systems engineering.

Privacy engineering can do better. Striving toward an enhanced uptake of privacy engineering in practice, this article highlights key aspects that need to be emphasized more prominently in the discourse. Drawing from lessons learned in various research projects and from extensive industry experience, we want to shed light on underrepresented, albeit crucial aspects of privacy engineering in the context of modern information systems engineering, thereby fostering its wide adoption in practice.

WHAT, THEN, IS PRIVACY ENGINEERING?

In our quest to unravel the core of privacy engineering, it becomes apparent that even the underlying concept of privacy is—like fairness, art, or democracy—an "essentially contested" one. We can basically agree on a term and its desirability, but its actual meaning is subject to a broad variety of different interpretations and inherently eludes reaching broad consensus on a single definition.[4] The same is true for privacy engineering. Conceptions range from the design and implementation of anonymity-preserving algorithms and protocols to higher-order ones taking up methods and practices from software engineering, physical architecture, human–computer interaction, or socio-technical systems design.

To this existing spectrum, we want to add another point of view that puts an explicit emphasis on practical applicability in real-world information systems. In particular, we look at privacy engineering from the perspective of enterprise information systems and architectures, established paradigms and practices for their development and operation in practice, and the associated requirements and constraints. By bringing these aspects into focus, we can identify and highlight the gaps that exist between the current state of the privacy engineering discourse and the prevailing practices within the realm of enterprise information systems.

BY BRINGING THESE ASPECTS INTO FOCUS, WE CAN IDENTIFY AND HIGHLIGHT THE GAPS THAT EXIST BETWEEN THE CURRENT STATE OF THE PRIVACY ENGINEERING DISCOURSE AND THE PREVAILING PRACTICES WITHIN THE REALM OF ENTERPRISE INFORMATION SYSTEMS.

This, in turn, allows us to identify aspects of crucial importance for privacy engineering to better align with real-world information systems engineering and, thus, to increase its practical relevance, applicability, and adoption. In this regard, we do in the following particularly highlight the needs to: 1. broaden the view beyond anonymization, data minimization and security; 2. consciously recognize what we call *second-order nonfunctional properties* of privacy mechanisms; and 3. relax on so far predominant "all-or-nothing" aspirations. 4. Finally, we also highlight how the provision of technical artifacts that are easily reusable in real-world environments can induce "indirected implementation obligations" and thereby foster the broad application of novel privacy mechanisms in practice.

BROADENING THE VIEW BEYOND ANONYMIZATION, DATA MINIMIZATION, AND SECURITY

While privacy engineering is often considered as merely an approach to implement anonymization and pseudonymization techniques or to ensure confidentiality,[5] privacy engineering entails a much broader range of goals and activities. Privacy-related regulations, such as the

GDPR or the CPRA, and nonregulatory frameworks, such as the Fair Information Practice Principles or the Organization for Economic Cooperation and Development Privacy Principles, clearly call for further principles to be properly reflected in the design and implementation of real-world information systems. These principles include:

- *Lawfulness (including legal basis such as consent):* The collection and processing of personal information has to be done in a lawful and fair manner. Under the GDPR and other privacy legislations, this can mean that any collection or processing of personal data is to be considered unlawful unless properly legitimized. Beyond individual consent, which is quite prominent in academic discussions, this legitimation can also rest on other legal bases, such as, the necessity for fulfilling a contract (think of address data being processed by an online shop) or legal obligations (e.g., an employer forwarding income data to tax authorities). Technical approaches for interlinking collection and processing of personal data with the respective underlying legitimation (allowing for subsequent reviews whether they are still valid, for instance) are, however, largely lacking.
- *Purpose limitation:* Slightly simplified, the principle of purpose limitation says that personal data are only to be processed for the purpose(s) they were initially collected for. For technically materializing this principle, information systems and the underlying data management solutions must allow for controlling the flow and use of personal data based on respective purposes and, thus, be "purpose-aware" by design. Approaches for, e.g., purpose-based access control will certainly prove valuable here.
- *Data minimization (including necessity):* Minimizing the amount of personal data being processed to what is absolutely necessary is what widespread "privacy" technologies for anonymization, pseudonymization, etc. are typically aimed at. It is worth noting that data minimization does not necessarily require minimizing the amount of data in general but only the amount of personal data. This can—albeit with some pitfalls—also be achieved by means of sufficiently reducing/removing the linkability between data and its subject. Similarly, in many cases, even a simple process for recording and maintaining data retention periods would already significantly limit the amount of personal data kept by many services in common use today.
- *Transparency:* To allow data subjects (users) to act and decide in a well-informed, self-sovereign manner, they must be provided with sufficient information on how their data is processed, for which purposes, etc. All of this information needs to be provided in a way that users can access and understand based on their individual abilities. Today, it is typically provided in textual privacy policies that are, however, barely legible by laypersons and more often than not conflict with today's well-established agile principles and practices of systems engineering. This is calling for more appropriate, technically mediated approaches and expecting industry to pick up state-of-the-art approaches, such as code scanning for personal information processing, utilizing application programming interfaces for communicating privacy policies of microservices, or alternative novel but mature transparency-by-design measures.
- *Security:* The traditional C-I-A triad (confidentiality, integrity, availability) of information security is also highly relevant in the context of privacy. Personal data need to be kept confidential and the integrity and availability of personal data are of crucial importance for avoiding any mistreatment (imagine, for example, unauthorized changes to or deletions of personal health records), as long as the data are actually relevant (while in case of irrelevance, the principle of data minimization would apply and explicitly call for deletion).
- *Accountability:* Like any other rule, privacy-related obligations would be rather meaningless without appropriate means for monitoring (or demonstrating) their fulfillment and for holding responsible parties accountable. With regard to privacy, this is traditionally achieved through a mix of technical and

nontechnical approaches, ranging from well-documented systems architectures over various technical mechanisms for trustworthy computing to in-depth on-site inspections by authorities and certification auditors. Under current givens of often cross-organizational processing of personal data in highly distributed and continuously changing information systems, however, these established means do hardly suffice to appropriately ensure accountability anymore.

Beyond these, further principles, such as data portability (allowing data subjects to transfer data from one service provider to another) or accuracy and fairness (ensuring that data are actually correct, not biased, and can be reviewed, corrected, or amended) may also be added to the set of relevant privacy principles that need to be reflected technically. Last but not least, nonregulatory conceptions of privacy also refer to similar principles that cannot be properly addressed by means of anonymization and security alone.

Instead of largely concentrating on ever new anonymization and security techniques, practice thus calls for a more encompassing set of functionalities covering all of the abovementioned principles. The technology scope of privacy engineering should thus be consciously broadened. Mapping the abovementioned principles to privacy-focused protection goals also including unlinkability, intervenability, and transparency (as, for instance, done in the "Standard Data Protection Model" proposed by German data protection authorities[6]) may also prove valuable here.

RECOGNIZING FUNCTIONAL AND NONFUNCTIONAL PROPERTIES OF PRIVACY MECHANISMS (AND ACKNOWLEDGING THE IMPORTANCE OF THE LATTER)

In information systems engineering, it is typically distinguished between functional and nonfunctional properties that systems have and respective requirements they must fulfill. Functional properties here refer to the core functionalities a system is meant to provide: a database stores and allows the querying of data or a travel planning service is able to calculate appropriate routes and travel times for different means of transportation. Nonfunctional properties or "qualities," in turn, refer to "constraint[s] on the manner in which [a] system implements and delivers its functionality."[7] Performance, scalability, or even security and privacy are typically mentioned here. Such nonfunctional properties are often crucial for the practical applicability or adoption of a technical system or component, irrespectively of its capacity to fulfill functional ones.

For privacy technologies, in turn, a similar differentiation must be made. From this perspective, functional properties refer to the privacy functionality a technical artifact provides: a certain property-preserving encryption scheme allows for a well-defined set of operations to be executed on encrypted data; a purpose-based access control scheme allows to technically enforce the privacy principle of purpose limitation, and so on. This is what we typically find in technical papers presenting novel privacy mechanisms, protocols, etc.

Nonfunctional properties of respective technical artifacts are, however, only rarely discussed. Nonetheless, these are of crucial importance for achieving applicability in practice. Based on existing research, we can identify at least the following nonfunctional properties of privacy mechanisms to be decisive for their practical application, albeit only rarely discussed in the privacy engineering literature:

- *(Re-)usability in relevant real-world information systems contexts:* One of the core requirements for privacy mechanisms to be actually adopted in practice is that they are provided as an easily (re-)usable artifact (e.g. library, package, component) that can be applied in conjunction with different systems of a particular class (e.g., different SQL databases) actually employed in practice.
- *Coherent integration into established software stacks, architectures, and development practices:* To foster practical adoption, a technical privacy mechanism must pay appropriate regard to the context it shall be applied in. A database extension with a modified query language, for instance, will hardly be applicable in conjunction with abstraction layers such as object-relational mappers widely used in practice. Development

paradigms and practices, such as agile DevOps, might also call for explicit recognition in the design of certain privacy mechanisms.[8] Aligning privacy engineering approaches with security practices, which are already much more mature and adopted in practice, would be another useful angle to ensure integration.

ALIGNING PRIVACY ENGINEERING APPROACHES WITH SECURITY PRACTICES, WHICH ARE ALREADY MUCH MORE MATURE AND ADOPTED IN PRACTICE, WOULD BE ANOTHER USEFUL ANGLE TO ENSURE INTEGRATION.

- *Developer-friendliness and low implementation efforts:* If a new technical privacy mechanism places a significant burden on the developers who shall apply or integrate it into their systems, this will hinder its adoption in a multitude of ways. Conversely, if applying a privacy mechanism merely requires minimal code modifications, developers will be much less reluctant. Similarly, management support also strongly depends on the implementation overheads that are to be expected.
- *Reasonable and experimentally determined performance overheads in realistic settings:* In many cases, the performance overhead raised by a novel privacy mechanism is rather unknown. In practice, however, the overhead to be expected is of crucial importance for deciding about a privacy mechanism's application. Explicitly provided overheads empirically gathered in experiments resembling real-world systems, environments, and workloads as closely as possible are therefore indispensable for making conscious and empirically well-founded decisions.

In the light of the abovementioned conception of privacy itself being a nonfunctional property of information systems, we refer to these properties of privacy mechanisms as *second-order nonfunctional properties*. These (and presumably additional ones) will foreseeably be decisive for a technical privacy artifact's actual transfer from its scientific birthplace into real-world applications. Nonetheless, they are only marginally present in the privacy engineering discourse.

LET PERFECTION NOT BE THE ENEMY OF THE GOOD

Another aspect quite prominent in the prevailing discourse regards the perceived need for solutions that provide some sort of formal guarantee that a given privacy property is 100% ensured in the light of a certain attacker model. Of course, technical mechanisms able to achieve this would always be the first choice, but in many cases, these come at the cost of significant drawbacks in matters of practical applicability. Mechanisms for fully homomorphic encryption or secure multiparty computation are a prime example here: In theory, they allow the outsourcing of critical calculations to external parties (such as cloud providers) while still providing confidentiality or integrity guarantees against these. However, such mechanisms usually come with tremendous performance overheads and lack easy integrability into real-world systems, hindering their application in practice. Similarly, adapted databases providing low-layer purpose-based access control have been proposed for materializing the principle of purpose limitation technically. However, these do not align with implementation stacks and data access models used in real-world information systems engineering, significantly limiting their practical applicability. Compared to these, alternative approaches for purpose-based access control explicitly aligned with such givens from practice while relaxing on aspects, such as circumventability by adversarial in-house developers[9] may turn out as the superior ones, given that they allow purpose-awareness to make it into real-world information systems at all. In matters of accountability, evidence doesn't need to be "provably unforgeable" to provide an actual benefit, and so forth.

By and large, it needs to be better recognized that regulations do not require the implementation of technical mechanisms that enforce privacy principles in a guarantee-like, 100% fashion. Instead, they follow a nonbinary, risk-based approach, calling for technical measures that properly reduce relevant risks (but not necessarily eliminate them completely

and provably). The GDPR, for instance, obligates data controllers to apply technical and organizational measures "designed to implement [privacy] principles" and explicitly links respective obligations to factors, such as the cost of implementation or the risks associated with the processing. From this perspective, an easy-to-implement, low-overhead mechanism that leaves a certain risk of circumvention by adversarial in-house developers can in many cases be preferable over one that provides formal guarantees, albeit at the cost of significant performance overheads.

In consequence, privacy engineering should, more often than currently, take a "realistic stance" on developers and data controllers. It must be weighed whether it is more important and valuable to support them in fulfilling their duties than trying to ensure absolute tamper- or concealment-proofness and end up without any mechanism being present at all.

CREATING IMPACT BY SHAPING THE STATE OF THE ART

One important question remains to be answered: How do we foster the actual adoption of privacy engineering in the industry? Privacy engineering and privacy-friendly systems will almost always lead to increased development and operational costs. Hence, beyond their need to comply with legislation, data controllers often have only limited incentives to make their systems more privacy-friendly than absolutely necessary.

Thus, if we aim to foster privacy engineering in practice, three interdependent main angles seem to be available: 1. increase user demand, 2. provide stricter and enforced obligations for industry, or 3. provide easy to use, feasible, and viable privacy-preserving technologies and methods. While addressing user demand is a topic we will not further consider here, the latter two approaches deserve more attention. Legislation already requires companies to apply privacy by design (e.g., Data Protection by Design and by Default in Article 25 GDPR). For multiple reasons, however, legislators usually refrain from stipulating specific technologies and methods. The actual technologies and methods to be used are to be derived from the state of the art, the risk caused by the processing, and other factors, such as cost. A cutting-edge technology raising serious integration or operational cost will therefore not be considered obligatory to apply in most cases. Thus, the key to increased adoption of privacy engineering methods, tools, and technologies lies to a large extent on the supply side and, therefore, in the provision of easily usable, effective and economically viable artifacts. Only on the basis of widespread availability and adoption of these artifacts will recognized industry practices emerge to form the state of the art to be considered by controllers.

The privacy engineering community's best avenues to advancing the practical adoption of privacy engineering, thus, lies in consciously advancing this state of the art. This requires several steps: First, we—in academia and industry—have to provide concrete, sufficiently mature, and publicly available implementations to demonstrate feasibility and effectiveness and to introduce the respective mechanism to the practice. Second, we must ensure that the implementation can be integrated into realistic information systems with low effort and high protection efficiency (see developer-friendliness and low implementation efforts in the "Recognizing Functional and Nonfunctional Properties of Privacy Mechanisms (and Acknowledging the Importance of the Latter)" section), and third, we must demonstrate economic feasibility, i.e., that operational overheads are reasonable (typically through, e.g., performance experiments with realistic scenarios and payloads).

BY AND LARGE, IT NEEDS TO BE BETTER RECOGNIZED THAT REGULATIONS DO NOT REQUIRE THE IMPLEMENTATION OF TECHNICAL MECHANISMS THAT ENFORCE PRIVACY PRINCIPLES IN A GUARANTEE-LIKE, 100% FASHION.

Together, these three factors may then, depending on the specific cost–risk assessment for a particular use case, imply an implicit regulatory expectation to implement a privacy mechanism in practice. This "obligation through implementation" approach can be consciously applied for fostering the actual adoption of novel technical privacy mechanisms in real-world information systems engineering.

Now that privacy engineering is gaining traction in industry, corporate heads of privacy engineering, chief information security officers, and their teams need to be empowered with proper technical tools and methods. For this to happen, privacy engineering needs to better align with real-world information systems engineering. In this article we have argued that this requires several things. At a more technical layer, privacy engineering needs to broaden its view beyond mere anonymization, data minimization, and security, and needs to properly address "second-order nonfunctional properties" of privacy mechanisms, like reusability or integration into established development practices. At a "policy" layer, it might be beneficial to abandon "all-or-nothing" approaches to privacy in some fields of academia to more easily bridge the gap between the academic world and industry. Regulators should continue striving toward risk-based approaches, while ensuring consistent, noncontradictory regulation. Companies, in turn, should seriously consider investing in their privacy engineering capabilities lest they find themselves lagging behind the state of the art by too far one day.

REFERENCES

1. L. Kissner and L. Cranor, "Privacy engineering superheroes," *Commun. ACM*, vol. 64, no. 11, pp. 23–25, 2021, doi: 10.1145/3486631.
2. K. Wuyts, L. Sion, and W. Joosen, "LINDDUN GO: A lightweight approach to privacy threat modeling," in *Proc. IEEE Eur. Symp. Secur. Privacy Workshops (EuroS&PW)*, Sep. 2020, pp. 302–309, doi: 10.1109/EuroSPW51379.2020.00047.
3. J. H. Hoepman, *Privacy Design Strategies (The Little Blue Book)*. 2018. [Online]. Available: https://www.cs.ru.nl/~jhh/publications/pds-booklet.pdf
4. D. K. Mulligan, C. Koopman, and N. Doty, "Privacy is an essentially contested concept: A multi-dimensional analytic for mapping privacy," *Philos. Trans. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 374, no. 2083, 2016, Art. no. 20160118, doi: 10.1098/rsta.2016.0118.
5. L. H. Iwaya, M. A. Babar, and A. Rashid, "Privacy engineering in the wild: Understanding the practitioners' mindset, organizational aspects, and current practices," *IEEE Trans. Softw. Eng.*, vol. 49, no. 9, pp. 4324–4348, Sep. 2023, doi: 10.1109/TSE.2023.3290237.
6. "The standard data protection model." BFDI. Accessed: Feb. 15, 2024. [Online]. Available: https://www.bfdi.bund.de/EN/Fachthemen/Inhalte/Technik/SDM.html
7. R. Taylor, N. Medvidovic, and E. Dashofy, *Software Architecture: Foundations, Theory, and Practice*. New York, NY, USA: Taylor & Francis, 2009, p. 447.
8. S. Gürses and J. Van Hoboken, "Privacy after the agile turn," Open Society Foundations, Peoria, IL, USA, May 2017. [Online]. Available: https://osf.io/preprints/socarxiv/9gy73
9. F. Pallas et al., "Towards application-layer purpose-based access control," in *Proc. 35th Annu. ACM Symp. Appl. Comput.*, 2020, pp. 1288–1296, doi: 10.1145/3341105.3375764.

FRANK PALLAS is senior researcher at the Information Systems Engineering Group of Technical University Berlin, 10587 Berlin, Germany. His research interests include privacy engineering and policy-aligned systems in real-world, enterprise-grade contexts. Pallas received a Ph.D. in computer science from Technical University Berlin. Contact him at frank.pallas@tu-berlin.de.

KATHARINA KOERNER is a corporate development manager with Daiki, San Jose, CA 95129 USA. Her research interests include tech policy, privacy, security, and artificial intelligence regulation. Koerner received a Ph.D. in European Law from Innsbruck University, Austria. Contact her at kk@dai.ki.

ISABEL BARBERÁ is a privacy engineer, artificial intelligence advisor, and cofounder of Rhite, 3526 KS Utrecht, The Netherlands. Her research interests include privacy and security engineering, artificial intelligence, and risk analysis, particularly implementation of threat modeling techniques. Barberá received a master's degree in philology and computational linguistics from the University of Murcia, Spain, and an advance master (LL.M) in law and digital technologies from the University of Leiden, The Netherlands. She is a member of the European Network and Information Security Agency Ad Hoc Working Group on Data Protection Engineering and a member of the European Data Protection Board Pool of Experts. Contact her at isabel@rhite.tech.

JAAP-HENK HOEPMAN is a guest professor at the PRISEC–Privacy And Security Group of Karlstad University,

651 88 Karlstad, Sweden; an associate professor at the Digital Security group of the Radboud University, 6500 HD Nijmegen, The Netherlands; and an associate professor in IT Law at the University of Groningen, 9712 CP Groningen, The Netherlands. His research interests include privacy by design and privacy friendly protocols. Hoepman received a Ph.D. in computer science from the University of Groningen, The Netherlands. Contact him at jhh@cs.ru.nl.

MEIKO JENSEN is a senior lecturer for cybersecurity and privacy at Karlstad University, 651 88 Karlstad, Sweden. His research interests include privacy engineering, cybersecurity, cloud security and privacy, data protection by design, anonymity and pseudonymity, and protection goals of privacy. Jensen received a Ph.D. in cloud security from Ruhr University Bochum, Germany. Contact him at Meiko.Jensen@kau.se.

NANDITA RAO NARLA is the head of technical privacy and governance at DoorDash, San Francisco, CA 94107 USA. Her research interests include privacy engineering, cybersecurity, and ethics in technology design and governance. Narla received an M.Sc. in information security from Carnegie Mellon University. She is a Senior Fellow at Future of Privacy Forum, and serves on the advisory boards and technical standards committees for the International Association of Privacy Professionals, Ethical Tech Project, X Reality Safety Initiative, Institute of Operational Privacy Design, and the National Institute of Standards and Technology. Contact her at nnarla@fpf.com.

NIKITA SAMARIN is a privacy researcher and a doctoral candidate in the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, Berkeley, CA 94709 USA. His research interests include the impact of existing software engineering practices on end-user privacy mobile systems security, biometric authentication, and secure machine learning. Samarin received a B.Sc. with honors in computer science from the University of Edinburgh. Contact him at nsamarin@berkeley.edu.

MAX-R. ULBRICHT is a technical officer for the Berlin Commissioner for Data Protection and Freedom of Information, 10555 Berlin, Germany. His research interests include privacy enhancing technologies for federated information systems. Ulbricht received a diploma in computer science from Technical University Berlin. Contact him at mru@meta-level.net.

ISABEL WAGNER is an associate professor in cybersecurity, Department of Mathematics and Computer Science, at the University of Basel, 4051 Basel, Switzerland. Her research interests include privacy and privacy-enhancing technologies, particularly metrics to quantify the effectiveness of privacy protection mechanisms, privacy protections for smart technologies, and measurement studies to create transparency for web and Internet of Things systems. Wagner received a Ph.D. in computer science from the University of Erlangen, Germany. She is a Senior Member of IEEE. Contact her at isabel.wagner@unibas.ch.

KIM WUYTS is a cyber and privacy manager at PriceWaterhouseCoopers Belgium, 1831 Diegem, Belgium. Her research interests include privacy engineering and application security, and bridging the gap between these two domains. Wuyts received a Ph.D. in privacy engineering from KU Leuven, Belgium. She is program cochair of the International Workshop on Privacy Engineering, and a member of European Network and Information Security Agency's working group on Data Protection Engineering. Contact her at kim.wuyts@pwc.com.

CHRISTIAN ZIMMERMANN is a senior expert for cybersecurity and privacy engineering at Robert Bosch GmbH, 70839 Gerlingen-Schillerhöhe, Germany. His research interests include automotive security and privacy, transparency-enhancing technologies, privacy economics and privacy-preserving technologies. Zimmermann received a Ph.D. in information systems from the University of Freiburg. He is a member of the European Network and Information Security Agency Ad Hoc Working Group on Data Protection Engineering. Contact him at christian.zimmermann3@de.bosch.com.

EDITOR: Doron Drusinsky, Naval Postgraduate School, ddrusins@nps.edu

DEPARTMENT: ALGORITHMS

This article originally appeared in
Computer
vol. 57, no. 7, 2024

From Concept to Reality: Leveraging Correctness-by-Construction for Better Algorithm Design

Tabea Bordis, Maximilian Kodetzki, and Ina Schaefer, *Karlsruhe Institute of Technology*

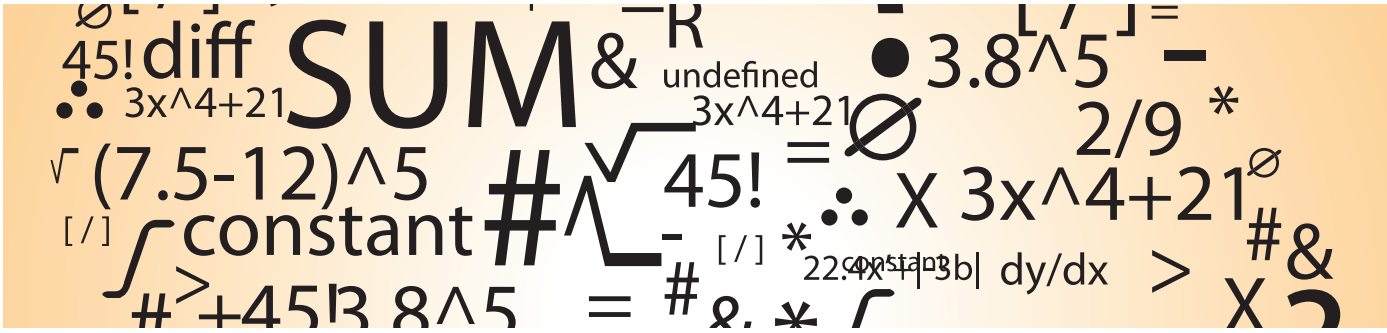
Algorithms demand both correctness and efficiency, but formal methods often lack support for ensuring these essential properties during algorithm construction. This article introduces correctness-by-construction (CbC) development which facilitates algorithm design through iterative refinement steps enabling the construction of correct and efficient algorithms.

Algorithms are an essential part of almost every software system. Building software systems can either involve reusing existing algorithms or developing new algorithms for specific problems. In each case, a good algorithm solves problems efficiently, but still in an understandable and maintainable way. Developing efficient algorithms, however, is a challenge even for experienced software engineers. First, efficiency is seldomly created using the most straightforward solution, but requires more thorough thinking. Second, the functional correctness of algorithms in libraries is of high importance since they are reused in a large community of users. Functional correctness of an algorithm can be shown using formal methods. Formal methods provide techniques and tools that generate stronger correctness guarantees than regular software testing. For example, deductive verification translates a program and its functional specification into mathematical proof obligations to establish functional correctness of the program. Just recently, the deductive verification tool KeY found a bug in the TimSort algorithm that is used by the Java standard library.[1] Most formal verification approaches are applied post hoc which means that an algorithm is only verified *after* it is implemented. Therefore, those post hoc approaches only support developers by showing functional correctness of their algorithms after development, but not during construction in the first place.

In this article, we demonstrate the incremental algorithm development approach *correctness-by-construction* (CbC) as imagined by Dijkstra,[2] Gries,[3] or Kourie and Watson,[4] where correctness is not an afterthought, but an integral part of the very construction process. CbC supports developers in constructing "better" algorithms which can refer to different properties of the constructed algorithm, such as correctness, efficiency, maintainability, or structural elegance. CbC is based on Hoare triples of the form {P} A {Q} consisting of a precondition P, an algorithm A, and a postcondition Q. Hoare triples represent correctness assertions that are true if and only if, starting from the precondition, the postcondition is satisfied after executing the algorithm A. The process of CbC has four steps. First, the pre- and postcondition are defined to create an abstract Hoare triple {P} S {Q}, where S is a placeholder for the algorithm that we want to create. Second, the placeholder is refined by applying one refinement rule from the predefined set of refinement rules. A refinement rule is a template to introduce common programming constructs, such as loops and selections, which defines an applicability condition to guarantee the correct preservation of the user-defined specification. A refinement rule might also insert more placeholders that still need to be refined. Third, the applicability condition of the applied refinement rule is checked. In this step, it might also be necessary to provide further specifications, such as loop invariants.

Digital Object Identifier 10.1109/MC.2024.3390948
Date of current version: 26 June 2024

 Published by the IEEE Computer Society



Fourth, if there is no placeholder left that still needs to be refined, the algorithm is complete and correct by construction. Otherwise, we repeat the process starting with the second step to refine the next placeholder to a program.

The underlying idea of this specification-first, refinement-based approach is that better algorithms can be constructed when the developer must think about their construction more thoroughly rather than hacking them into correctness. As a result, when applying CbC compared to classical post hoc verification, errors are more likely to be detected earlier in the design process, and the structure of the algorithm tends to be clearer resulting in more elegant and efficient solutions.[4] To further spread correct-by-construction software development, we implemented CbC in the CorC ecosystem.[5] CorC is a graphical and textual integrated development environment (IDE) to construct algorithms following the CbC approach. CorC supports developers to refine their program by applying refinement rules and to verify the correct application of these refinement rules using the deductive verification tool KeY.[6] Evaluation results even show a decreased verification effort compared to post hoc verification.[7]

In the following, we introduce the CbC methodology and highlight its benefits for constructing algorithms using an example. Additionally, we give an overview of the CorC ecosystem to demonstrate how our research on extending the applicability of CbC meets the challenges of developing today's software systems. We present four lines of research where CbC is integrated into software engineering processes to scale its applicability from single algorithms to object-oriented and component-based software systems and used beyond functional correctness.

{P} S {Q}	*can be refined to*
1. Skip :	{P} *skip* {Q} *iff* P *implies* Q
2. Assignment :	{P} x := E {Q} *iff* P *implies* Q[x\E]
3. Composition :	{P} S_1 ; S_2 {Q}
	iff there is an intermediate condition M *such that* {P} S_1 {M} *and* {M} S_2 {Q}
4. Selection :	{P} **if** $G_1 \rightarrow S_1$ **elif** ... $G_n \rightarrow S_n$ **fi** {Q}
	iff (P *implies* $G_1 \vee G_2 \vee \ldots \vee G_n$) *and* {P ∧ G_i} S_i {Q} *holds for all* i.
5. Repetition :	{P} **do** [I, V] G → S **od** {Q}
	iff (P *implies* I) *and* (I ∧ ¬G *implies* Q) *and* {I ∧ G} S {I} *and*
	{I ∧ G ∧ V=V_0} S {I ∧ 0≤V ∧ V<V_0}

FIGURE 1. Set of CbC refinement rules.[4]

CORRECT-BY-CONTRUCTION ALGORITHM DEVELOPMENT

CbC[4] is a refinement-based, incremental approach to develop algorithms in the sense of total correctness. Every statement of the algorithm is surrounded by a specification forming a Hoare triple of the form {P} S {Q}. Thereby, the precondition P marks the state of the program before the statement S is executed and guarantees that the statement will terminate in the state described by postcondition Q. Precondition P, postcondition Q, and the Hoare triple itself are predicate formulas evaluating to true or false. The pre- and postconditions are expressed in first-order logic and the statements in Guarded Command Language,[8] using the following five constructs: empty command (*skip*), assignment (:=), composition (;), selection (**if**), and repetition (**do**). In Figure 1, we show the set of refinement rules that are used to refine an abstract statement S to a concrete statement in Guarded Command Language. Each of the refinement rules contains an applicability condition that has to be fulfilled to guarantee the correctness of the refinement with respect to the specification.

- *Skip*: Skip statements do not alter the program state.
- *Assignment*: An abstract statement S can be refined to an assignment x: = E, if precondition P

EACH OF THE REFINEMENT RULES CONTAINS AN APPLICABILITY CONDITION THAT HAS TO BE FULFILLED TO GUARANTEE THE CORRECTNESS OF THE REFINEMENT WITH RESPECT TO THE SPECIFICATION.

implies postcondition Q in which the variable *x* has been replaced by expression E, noted as Q[*x*\E].

- *Composition*: The composition rule splits one abstract statement S into two abstract statements S_1 and S_2 that are executed sequentially. Additionally, an intermediate condition M has to be provided.
- *Selection*: The selection rule branches the abstract statement into different cases that are defined by the guards G_i. The Hoare triple is refined to *n* more Hoare triples of the form $\{G_i \wedge P\}\ S_i\ \{Q\}$. The substatement of the first satisfied guard G_i is executed.
- *Repetition*: The repetition rule introduces a loop that executes statement S as long as guard G evaluates to true. The repetition refinement rule additionally requires an invariant I and a variant V. To verify termination of the loop, it is checked that the variant is monotonically decreasing with zero as the lower bound. Additionally, an invariant is needed to guarantee the postcondition.

DUTCH NATIONAL FLAG ALGORITHM

In the following, we develop an algorithm using CbC to solve the Dutch National Flag problem that has been proposed by Dijkstra in 1976. The Dutch National Flag problem is a special sorting problem that considers an array with three different entries (in our case the colors of the Dutch National Flag: red, white, and blue). This array shall be sorted into the correct order regarding the colors of the Dutch National Flag. While one naive solution would be to loop through the array twice (once for counting the number of red, white, and blue entries and a second time to assign the correct values to the entries of the array), there is a more efficient solution to this problem that only uses one loop. We show how the refinement-based approach of CbC is applied to construct an algorithm for this problem and highlight how this specification-centric process helps to develop a more efficient solution compared to the naive approach.[4] This means that we do not only present the final solution with the final specifications, but we mimic the thought process of developing the algorithm for the very first time.

Before we start developing the algorithm, we define two predicates that help us to specify the problem concisely. We define the predicates intuitively, as a developer would do when defining the problem for the first time. The first predicate, color*(A, l, h, x),* evaluates to true if and only if the entries of the array *A* from index *l* to index *h* have the color *x*. The second predicate sorted*(A, l, h, wb, wt, bb)* defines our sortedness criterion, which means that first in the array we have the red entries, followed by blue and white entries. The integers *wb*, *wt*, and *bb* (for white bottom, white top, and blue bottom) define the borders between the red, white, and blue entries in the array.

Definition 1: Predicate color:

color(Array A,int *l*,int *h*,value *x*) $\triangleq$
$\forall$ int *i*: $(l \leq i < h) \rightarrow A[i] = x$

Definition 2: Predicate sorted:

sorted(Array *A*,int *l*,int *h*,int *wb*,int *wt*,
int *bb*) $\triangleq$ color(*A*,*l*,*wb*,red) $\wedge$ color
(*A*,*wb*,*wt*,white) $\wedge$ color(*A*,*bb*,*h*,blue)
$\wedge$ $(0 \leq l \leq wb \leq wt \leq bb \leq h \leq A.len)$.

The first step of CbC is the definition of the problem in form of a Hoare triple specification {P} S {Q}. Since our predicate sorted describes the general form of the array, we can instantiate the bounds accordingly to express 1) an unsorted state of the array in the precondition and 2) a completely sorted state of the array in the postcondition. For the postcondition, we additionally define that *wt* has to be equal to *bb* since blue and white are neighboring colors. (Side note: For brevity, we omit defining some properties, which do not directly influence the development of this algorithm. For example, we could explicitly require the array to only have entries of the colors red, white, or blue in the precondition. Additionally, sorting algorithms should always be specified with a permutation property, such that solutions where the array simply is overwritten

with one value are not sufficient.) Formally, we define the starting Hoare triple as follows:

Definition 3: Problem:
{sorted(*A*,0,*A.len*,0,0,*A.len*)} *S*
{sorted(*A*,0,*A.len*,*wb*,*wt*,*bb*) ∧
(*wt* = *bb*)}

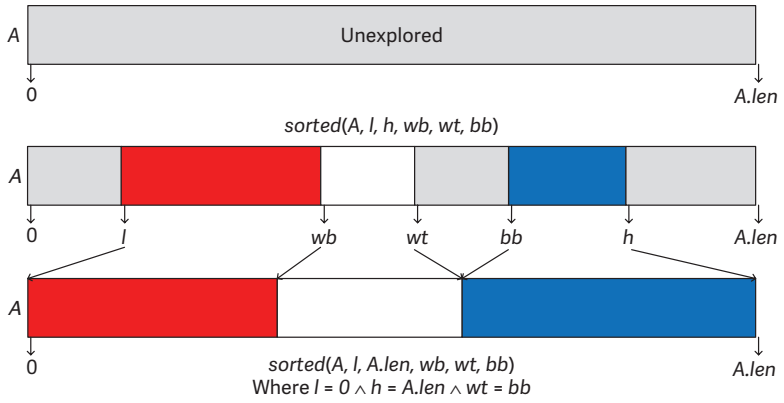


FIGURE 2. Illustration of Dutch National Flag algorithm (adapted from Kourie and Watson[4]).

In Figure 2, we illustrate the process of the intended Dutch National Flag algorithm with a starting, intermediate, and end state using predicate sorted. The main part of the algorithm will be a loop to sort the entries for which we have to define a loop invariant. A loop invariant describes a property that is true before and after every loop iteration. For our algorithm, it must reflect an intermediate state as shown in Figure 2. By examining Figure 2 during the definition of the loop invariant, we now notice that we can simplify our first, intuitively chosen specification. That is, there is no real need to have three unexplored regions in an intermediate state; one is sufficient. Therefore, we fix *l* and *h* to 0 and *A.len,* respectively, inducing an unsorted region between white and blue. We formally define the invariant using predicate sorted as follows:

Definition 4: Loop invariant:
Inv ≙ sorted(*A*,0,*A.len*,*wb*,*wt*,*bb*)

At this point, we notice that our definition of predicate sorted can be simplified since *l* is always 0 and *h* is always *A.len*. We revise our definition for predicate sorted in the problem and in the invariant accordingly as follows:

Definition 5: Predicate sorted (revised):
sorted(Array *A*,int *wb*,int *wt*,int *bb*)
≙ color(*A*,0,*wb*,red) ∧ *color*
(A,*wb*,*wt*,white) ∧color(*A*,*bb*,*A.len*,blue)
∧ (0 ≤ *wb* ≤*wt* ≤ *bb* ≤ *A.len*)

Definition 6: Problem (revised):
{sorted(*A*,0,0,*A.len*)} *S*
{sorted(*A*,*wb*,*wt*,*bb*) ∧ (*wt* = *bb*)}

Definition 7: Loop Invariant (revised):
Inv ≙ sorted(*A*,*wb*,*wt*,*bb*)

After defining the specification, we can start applying refinement rules to construct the Dutch National Flag algorithm. While it might seem like a lot of work for a software engineer before even starting the development, we want to highlight that we already gained a lot of insights about the Dutch National Flag problem and revising the problem description helps us to find a simple solution during program construction. Additionally, when developing an algorithm for a new problem, the iterative process is usual.

In Figure 3, we show all refinement steps that we applied to construct the Dutch National Flag algorithm in a refinement tree structure. The single refinement steps are numbered in application order with circled numbers and the name of the applied CbC refinement rule (see Figure 1). The box in the top right corner lists all conditions that are used as specifications throughout the construction process. The root node of the refinement tree is the starting triple with our previously defined pre- and postconditions. We already know that the main part of the algorithm will be a loop. Before entering the loop, we need to initialize our index variables *wt*, *wb*, and *bb* appropriately. Therefore, we split the abstract statement S into two sequentially executed abstract statements S1 and S2 using the composition refinement rule in refinement step 1. The applicability condition for the composition refinement

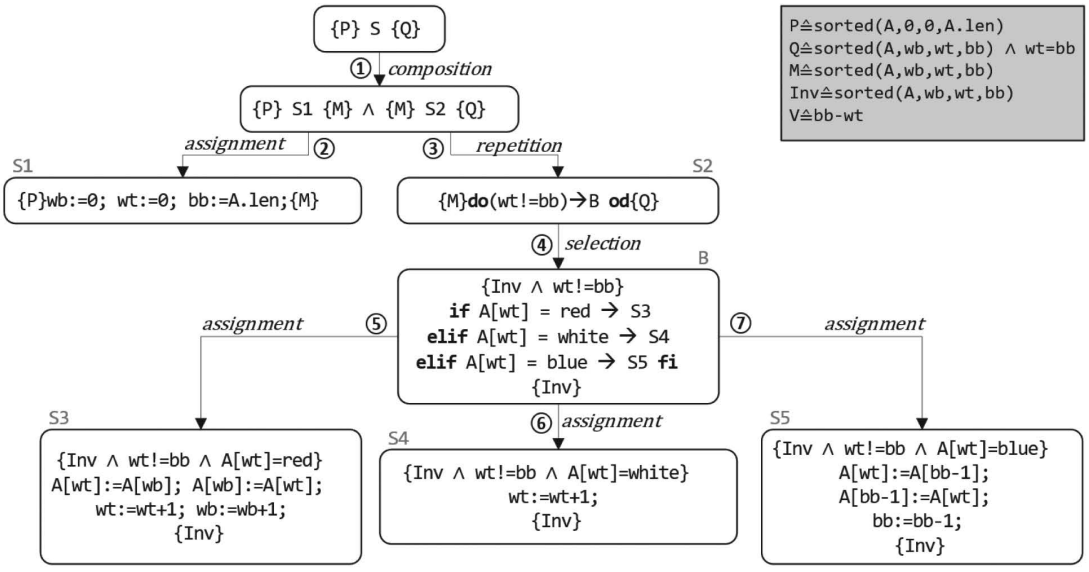


FIGURE 3. Dutch National Flag algorithm as CbC refinement tree.

rule requires us to define an intermediate condition M such that the two newly constructed Hoare triples {P} S1 {M} and {M} S2 {Q} hold. We define M to be the loop invariant *Inv* (see Definition 7) since it should hold after initializing the index variables and before entering the loop. The correctness of this refinement step is determined by checking the correctness of the refinement rules that are applied to refine S1 and S2.

In the second refinement step, we refine abstract statement S1 with the assignment refinement rule to initialize the index variables *wb*, *wt*, and *bb*. The applicability condition of the assignment rule checks that the precondition implies the postcondition of the triple, that is the loop invariant, after setting the values of the assignment as follows:

$$\text{sorted}(A,0,0,A.len) \rightarrow \text{sorted}(A,wb,wt,bb)[wb\backslash 0, wt\backslash 0, bb\backslash A.len]$$
$$\equiv \text{sorted}(A,0,0,A.len) \rightarrow \text{sorted}(A,0,0,A.len) \equiv \text{true}.$$

In the third refinement step, we refine abstract statement S2 with the repetition refinement rule. For this rule, we already defined the loop invariant *Inv* (see Definition 7), but still need to define a variant *V* and a loop guard. In our case, the loop is terminated when the unsorted region between indices *wt* and *bb* is empty, that is, *wt* = *bb*. The variant is needed to prove termination of the loop. It is defined such that it decreases monotonically with every iteration and has 0 as lower bound. We set the variant *V* to (*bb* – *wt*) since this number decreases steadily because in every iteration, we sort one element from the unsorted region between indices *wt* and *bb* either to the region before *wt* (increasing *wt* by one) or to the region after *bb* (decreasing *bb* by one). With that, we can establish the applicability conditions. For brevity, we do not give details on all applicability conditions. The interested reader can read the corresponding chapter in Kourie and Watson, 2012.[4]

In the fourth refinement step, we use the selection refinement rule to refine the abstract loop body B. We want to distinguish between cases where the next element we are looking at is either red, blue, or white and set the guards accordingly. The applicability condition of the selection refinement rule also requires us to define the guards in a way that always one of the guards is fulfilled. To prove this condition, we would in fact need the additional condition that our array cannot contain any other element than those three colors which are omitted for brevity as mentioned earlier. Note, that we did not define a classic loop iteration index *i*, as done for many loops. This is because index *i* would loop from 0 to the start of color blue, and therefore we noticed that index *i* and *wt* are increased at the same time and we can just use *wt* instead of *i*. Another thing to notice is that the specification for the

loop body B has not directly been inherited as we have seen for the assignment refinement rule in step 2. This is defined in the applicability condition of the repetition refinement rule and makes sure that the loop body fulfills the invariant *Inv* before and after execution and also includes the guard *wt!=bb* in the precondition.

The last refinement steps (steps 5–7) refine the abstract statements S3–S5 for the single cases of the selection statement using the assignment refinement rule. The trick here is to swap the element at index *wt* into the correct area and to increase (in case of *wt* and *wb*) or decrease (in case of *bb*) the corresponding indices afterward.

THE CorC ECOSYSTEM

We extended CbC to different fields of application that benefit from the idea of a structured development process guided by specifications and refinements. We categorize our research in two goals: The first goal is to develop concepts and tool support that scale CbC from developing single algorithms to whole software systems with the complexity of today's software. The second goal is to give guarantees beyond functional correctness for the constructed algorithms. All of our lines of research are combined in one open source tool: the CorC ecosystem. The core of the CorC ecosystem is the tool CorC[7] which is an Eclipse plug-in supporting the development of programs with CbC. CorC comes with a graphical and a textual editor. The graphical editor visualizes the CbC program in a tree structure, similar to Figure 3, such that bugs in the construction of the algorithm can be easily traced. The textual editor is implemented using a grammar for CbC programs. The beginning of a CbC program is a Hoare triple, which can then be refined by applying CbC refinement rules per drag-and-drop. In the background, the deductive verification tool KeY[6] is used to prove the applicability condition of each refinement rule. The following lines of research each extend the core functionality of CorC following our aforementioned goals. Further details are provided in the referenced papers and on GitHub.[9]

Object-oriented development using CbC

To enable widespread application of CbC for algorithms, we integrated object-oriented programming, as widely used programming paradigm, into CorC and improved the development process to enable the development using CbC alongside other verification strategies or classical testing.[10] Object-oriented programming introduces classes with fields and class invariants to CbC, which allows to develop more complex projects including inheritance and interfaces. Our roundtrip engineering process facilitates seamless transition from existing or manually written code to CbC development to correct code and vice versa. We integrated these concepts along with further usability features that simplify the development using CbC in the successor of CorC, called CorC 2.0.

WE WANT TO DISTINGUISH BETWEEN CASES WHERE THE NEXT ELEMENT WE ARE LOOKING AT IS EITHER RED, BLUE, OR WHITE AND SET THE GUARDS ACCORDINGLY.

CbC software architectures

Component-based architectures allow to establish a set of modular, reusable, and correct-by-construction components. This is equally interesting for libraries, where implementations are accessed through interfaces, and for third-party developments that are easier to integrate into individual projects. Most importantly, creating components that modularize correct implementations allows developers to think about how to compose software systems instead of how to program a monolithic software system from scratch. We argue that this is the foundation for building large and complex systems that take advantage from a CbC-based development style. Our extension, ArchiCorC, connects Unified Modeling Language-style component modeling, formal specification, and code generation, facilitating the creation of correct-by-construction components and their seamless integration into software systems.[11]

CbC for software product lines

Software product lines offer systematic reuse paired with variability mechanisms to realize whole product families. The commonalities and differences of the product variants are communicated as features, whose relationships are often modeled in feature

models. Guaranteeing the correctness of a product line is challenging, especially due to the number of possible product variants resulting from the number of feature configurations and the variable code structures. To create a correct product line using CbC, we extended the original CbC approach with a new refinement rule for variability mechanisms.[12] Additionally, we combined this mechanism with contract composition for variability in the pre- and postcondition.[13] We call this extension *variational CbC*.[14] The corresponding extension of CorC, VarCorC, uses FeatureIDE[15] and variational CbC to support the development of correct-by-construction software product lines.

BEYOND FUNCTIONAL CORRECTNESS: X-BY-CONSTRUCTION

Driven by our research in the past years on fields of application and extensions of the CorC ecosystem as a tool for developing correct-by-construction programs, we can see our vision of scaling CbC as necessary practice in modern software engineering coming together. Underlined by the participants of multiple user studies, who attest CbC's ability to effectively develop correct code, we believe that CbC, coupled with advanced tool support, is the go-to paradigm for functionally correct engineered software.[16,17] However, functional correctness is no longer the only criteria by which the quality of algorithms is measured. Nonfunctional properties play an increasingly important role in today's fast-paced, optimized world. Covered by the term *X-by-construction*, we do research on the refinement-based development of algorithms that fulfill certain nonfunctional properties by construction. We envision a fully comprehensive programming paradigm in which both the functional correctness of algorithms as well as the fulfillment of nonfunctional properties, such as security, resource consumption, and reliability, can be ensured using the by-construction approach.

Security-by-construction

Besides functional correctness, it is also important to consider security in program development. To express confidentiality and integrity of data, an information flow policy can be used to define how information may flow in a program (for example, a flow from public to secret data is allowed, but the other way is prohibited to ensure confidentiality and integrity of the data). Our extension of CbC to ensure this type of security-by-design is called *information flow control-by-construction* (IFbC).[18] Programs are constructed incrementally using refinement rules to follow an information flow policy. In every refinement step, security and functional correctness of the program is guaranteed, such that insecure programs are prohibited by construction. IFbC is implemented in an extension of CorC. We envision enabling security-by-design by extending IFbC with quantitative and probabilistic information flow specifications.

X-by-construction: Next steps

Induced by the climate crisis and rising energy prices, sustainable and *green* software has become an increasing focus in software engineering. Since the 2000s, research in the field of efficient software has grown steadily. *Cost analysis* has emerged as an important aspect in software development, where the efficiency of code is usually determined post hoc, that is, after the implementation of an algorithm, using various consumption parameters (for example, number of instructions executed, amount of execution time, memory allocated). Our current research aims to apply existing approaches to the by-construction paradigm. The goal is to specify efficiency-influencing parameters before implementing software and then to incrementally develop code that complies with the specification. Doing so, the efficiency of algorithms is to be determined by a specification before starting the development of code and guaranteed to be met after the implementation.

Besides resource consumption, our aim is to investigate how a by-construction approach can guarantee programming principles, such as robustness, resilience, and reliability. The challenge here lies in the lack of a calculus to formally specify and reason about those properties in the first place.

Correctness-by-construction development supports the construction of provably correct and efficient algorithms already in the design phase. We presented the incremental refinement-based process at the example of the Dutch National Flag problem as well as the tool CorC, that brings CbC into practice. With extensions for object-oriented programs and software-intensive systems, the CorC ecosystem covers common areas of today's software engineering.

Our current research aims to extend CbC to non-functional properties such as security, reliability, and resource consumption.

ACKNOWLEDGMENT

This work was supported in part by funding from the pilot program Core-Informatics of the Helmholtz Association (HGF). The authors would like to thank Loek Cleophas, Bruce W. Watson, und Derrick G. Kourie for longstanding collaboration on advocating correctness-by-construction engineering.

REFERENCES

1. S. de Gouw, F. S. de Boer, R. Bubel, R. Hähnle, J. Rot, and D. Steinhöfel, "Verifying OpenJDK's sort method for generic collections," *J. Automated Reasoning*, vol. 62, no. 1, pp. 93–126, 2019, doi: 10.1007/s10817-017-9426-4.
2. E. W. Dijkstra, *A Discipline of Programming*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1976.
3. D. Gries, *The Science of Programming*. New York, NY, USA: Springer-Verlag, 1981.
4. D. G. Kourie and B. W. Watson, *The Correctness-by-Construction Approach to Programming*. Heidelberg, Germany: Springer-Verlag, 2012.
5. T. Bordis, T. Runge, A. Kittelmann, and I. Schaefer, "Correctness-by-construction: An overview of the CorC ecosystem," *ACM SIGAda Ada Lett.*, vol. 42, no. 2, pp. 75–78, 2023, doi: 10.1145/3591335.3591343.
6. W. Ahrendt, B. Beckert, R. Bubel, R. Hähnle, P. H. Schmitt, and M. Ulbrich, *Deductive Software Verification—The KeY Book*. Cham, Switzerland: Springer-Verlag, 2016.
7. T. Runge, I. Schaefer, L. Cleophas, T. Thüm, D. Kourie, and B. W. Watson, "Tool support for correctness-by-construction," in *Proc. Int. Conf. Fundam. Approaches Softw. Eng.*, Cham, Switzerland: Springer-Verlag, 2019, pp. 25–42.
8. E. W. Dijkstra, "Guarded commands, nondeterminacy and formal derivation of programs," *Commun. ACM*, vol. 18, no. 8, pp. 453–457, 1975, doi: 10.1145/360933.360975.
9. "CorC Repository." GitHub. Accessed: March 28, 2024. [Online]. Available: https://github.com/KIT-TVA/CorC
10. T. Bordis, L. Cleophas, A. Kittelmann T. Runge, and B. W. Watson, "Re-CorC-ing KeY: Correct-by-construction software development based on KeY," in *The Logic of Software: A Tasting Menu of Formal Methods*. Cham, Switzerland: Springer-Verlag, 2022, pp. 80–104.
11. A. Knüppel, T. Runge, and I. Schaefer, "Scaling correctness-by-construction," in *Proc. Int. Symp. Leveraging Appl. Formal Methods*, Heidelberg, Germany: Springer-Verlag, 2020, pp. 187–207.
12. T. Bordis, T. Runge, and I. Schaefer, "Correctness-by-construction for feature-oriented software product lines," in *Proc. Int. Conf. Generative Program., Concepts Experiences*, 2020, pp. 22–34, doi: 10.1145/3425898.3426959.
13. T. Bordis, T. Runge, D. Schultz, and I. Schaefer, "Family-based and product-based development of correct-by-construction software product lines," *J. Comput. Lang.*, vol. 70, Jun. 2022, Art. no. 101119, doi: 10.1016/j.cola.2022.101119.
14. T. Bordis, T. Runge, A. Knüppel, T. Thüm, and I. Schaefer, "Variational correctness-by-construction," in *Proc. 14th Int. Working Conf. Variability Model. Softw.-Intensive Syst.*, 2020, pp. 1–9, doi: 10.1145/3377024.3377038.
15. T. Thüm, C. Kästner, F. Benduhn, J. Meinecke, G. Saake, and T. Leich, "FeatureIDE: An extensible framework for feature-oriented software development," *Sci. Comput. Program.*, vol. 79, pp. 70–85, Jan. 2014, doi: 10.1016/j.scico.2012.06.002.
16. T. Runge, T. Bordis, T. Thüm, and I. Schaefer, "Teaching correctness-by-construction and post-hoc verification—The online experience," in *Proc. Formal Methods Teaching Workshop*, Cham, Switzerland: Springer-Verlag, 2021, pp. 101–116.
17. T. Runge, T. Thüm, L. Cleophas, I. Schaefer, and B. W. Watson, "Comparing correctness-by-construction with posthoc verification—A qualitative user study," in *Refine*. Cham, Switzerland: Springer-Verlag, 2019, pp. 388–405.
18. T. Runge, A. Knüppel, T. Thüm, and I. Schaefer, "Lattice-based information flow control-by-construction for security-by-design," in *Proc. 8th Int. Conf. Formal Methods Softw. Eng.*, 2020, pp. 44–54, doi: 10.1145/3372020.3391565.

TABEA BORDIS is a researcher at Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany. Contact her at tabea.bordis@kit.edu.

MAXIMILIAN KODETZKI is a researcher at Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany. Contact him at kodetzki@kit.edu.

INA SCHAEFER is a professor of software engineering at Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany. Contact her at ina.schaefer@kit.edu.

EDITORS: Norita Ahmad, American University of Sharjah, nahmad@aus.edu
Preeti Chauhan, IEEE Reliability Society, preeti.chauhan@ieee.org

This article originally appeared in
Computer
vol. 56, no. 6, 2023

DEPARTMENT: DATA

Data's Impact on Algorithmic Bias

Donghee Shin and Emily Y. Shin, *Zayed University*

Algorithmic bias refers to systematic and structured errors in an artificial intelligence system that generate unfair results and inequalities. This column discusses how bias in algorithms appears, amplifies over time, and shapes people's thinking, potentially leading to discrimination.

Algorithms are human artifacts in that they are made, designed, trained, and applied by humans. Contrary to popular beliefs, AI is neither objective nor fair.[1]

WHY IS AI VULNERABLE TO BIAS?

An algorithm's performance largely hinges on people who designed them, the code they used, the data they analyzed for the machine learning (ML) models, and the way they trained the models. Algorithms are human made; therefore, they are naturally at risk of an inherent bias or specifically, cognitive bias, a pattern of deviation from rationality in judgment. This bias often leads to misrepresentation, wrong judgment, or misinterpretation. Humans create their own individual social reality more from their perceived world of reality and less from the objective input from it. Humans' cognitive bias are often transpired into algorithms they design, and thus AI produces and amplifies the bias that humans entered. For AI to be aligned with human preferences, it must learn those preferences. Learning human values by ML carries inherent risks. The underlying cause for AI bias lies in the conscious or unconscious human bigotry embedded in it throughout the development of algorithms. Hence, biases are quietly encoded during the process of algorithm creation.

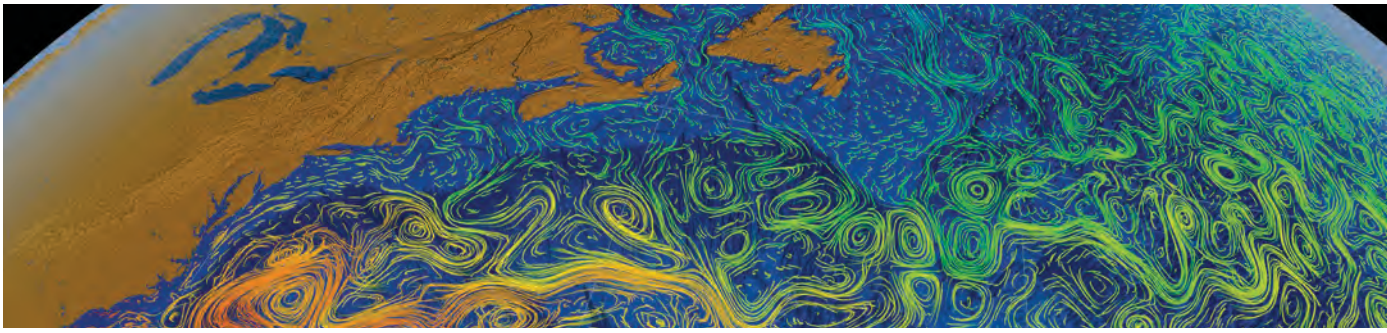
Algorithms are programmed by people who, even with goodwill, can be prejudiced and discriminate within an unfair social world; thus, algorithms reflect and amplify the larger prejudices of reality.[2] This type of amplification, which is called *algorithmic amplification* is common in the platforms with which we interact every day because platforms such as Google, TikTok, Instagram, and YouTube are designed toward organized data gathering, automated processing, distribution, and maximizing monetization of customer data. These platforms use vast amounts of data for algorithmic systems and have far expanded their capacities in what they can do to drive people to decisions and behaviors that maximize monetization.

The history of people's likes, clicks, comments, and retweets is the data that power the algorithmic amplification. Some social communities benefit more from algorithmic amplification than others. This is a reality on most of the platforms we use nowadays. The history of our likes, shares, and comments are the data driving the algorithmic amplification.

Herbert Simon[3] presented the idea of bounded rationality, which is limited by imperfect information, cognitive ability, and time constraints. Bounded rationality plays a key part in the way humans design algorithms and in the way automation bias comes from. YouTube's recommendations amplify sensational content to increase the number of people on the screen as well as the duration for which they are on the screen. Platforms recommend videos concerning political information, inappropriate content, and hate speech as ways to maximize revenue. News recommender systems suggest either negative or incorrect information—likely to provoke rage. Fake news headlines are designed to be exaggerated news rather than real facts.[4] Programmers do the coding of the algorithms, markets choose the data used by the algorithms, and the algorithm designers decide how

Digital Object Identifier 10.1109/MC.2023.3262909
Date of current version: 31 May 2023

 Published by the IEEE Computer Society



to apply the results of the algorithms. If the data analyzed by algorithms or used to train ML do not reflect the various parameters of users correctly, the results would be biased. Bias can enter algorithms and ML because of preestablished social, cultural, and political inequalities in society, and thus people, which can impact decisions regarding how data are gathered, filtered, coded, or selectively analyzed to frame ML. It is easy for people to let biases enter, which AI then algorithmizes and automates. That is why AI often makes decisions that are systematically unfair to certain groups of people.

Algorithmic bias is a set of systematic faults in algorithmic systems that generate unfair discriminations, such as favoring a certain group of users over others.[5] A bias can be either intended or unintended, and it can emerge from a misunderstanding or a misinterpretation, that is, the intentional design of certain algorithms or unexpected decisions associated with the way data are gathered, analyzed, or included to train ML. Relevant inquiries have found that these biases can potentially cause significant harm to the public.[6] A study at Harvard University revealed that AI-driven speech recognition systems show significant racial disparities, with voice recognition misunderstanding 40% of words from minority users and only 11% of those from white users.[7] Algorithmic bias has commonly been found in social media platforms and search engine results. This bias can have serious effects on intensifying social stereotyping, biases, and prejudice. Algorithmic bias is prevalent in every aspect of our lives, as biased algorithms are embedded throughout health care, criminal justice, and employment systems, influencing critical decisions, operational work processes, and working rules. ML helps technologies understand human rhetoric, bias, and discourse and has been found to reflect gender, racial, and class inequalities. Human biases, such as stereotyped sentiments attached to certain races, high-salary professions linked to a specific gender and race, and negative imaging of certain sexual orientations, become popularized to a wide variety of services.

HOW DO SUCH BIASES ENTER A SET OF ALGORITHMS?

Humans write the codes in algorithms, select the data used by algorithms, and decide how to present the results of the algorithms. As humans develop an algorithmic structure, human biases inevitably are written into the algorithms. Biases are implanted through algorithmic data, ML embeds these biases, and AI reflects these biases in their performance.[8] Algorithms themselves also contribute to biases. AI systems do not process and generate results only based on user data. They can also operate self-learning and self-programming algorithms based on secondary data, nonobservational, and situational data such as synthetized data, simulations, bootstrapped data, or a combination of generalized assumptions or rules. ML processes such data and learns from the data. People whose data was not processed or who have not otherwise been taken into consideration may also be directly involved and negatively impacted, particularly when algorithmic systems are used to inform critical decision-making. "(Algorithms) are embedded within larger social systems and processes, inscribed with the rules, values and interests of a typically dominant group."[9]

Algorithms reveal glimpses into the existing structure of bias and inequalities that are embedded within our social, economic, and political systems. Without conscientious and rigorous mental investigations, it is easy for humans to input human biases intentionally and unintentionally into algorithms. Then, the biases are amplified, regenerated, and propagated.

Algorithmic bias can be seen across various platforms. Social media platforms that contain biased algorithms exacerbate misinformation, fake news, and disinformation.[10] There has been fake news accusing racial groups, illegal immigrants, and even governments of the diffusion of COVID-19. Certain

political groups propagate fake news for the sake of political gains. Misinformation and disinformation about political campaigns harm democracy because people lose trust in the political system.[5] The real threat of algorithms is that they can amplify and magnify biases that already exist in the world. Realizing the seriousness of the bias issue, most firms have started to run programs to fight against bias and societal inequalities. For example, Amazon operates thorough antidiscrimination policies, recruits diverse

THIS TASK SHOULD BE DONE BY HUMAN FACT-CHECKERS BY SEARCHING FOR SOCIAL MEDIA POSTS OR ONLINE INFORMATION WITH SIMILAR QUERIES AND INFORMATION.

races, trains to recognize potential employee bias, and promotes diversity. All these efforts may be in vain if the AI models continue to operate in routine operations and the delivery of results remains inadvertently discriminating. Removing or mitigating algorithmic bias involves efforts beyond technical fixes. The tools and methods used to remove bias and reduce variance tend to cause another bias. Removing algorithmic bias should involve not only changing the algorithms or the systems but also changing cultural biases and social structures. Bias can perpetuate algorithmic inner systems because of preestablished social and cultural values. Society should continuously request that critical decisions be transparent, fair, and accountable, even as they become more and more algorithmized.

A NEGATIVE FEEDBACK LOOP AND BIAS

A feedback loop is part of a system in which some parts of the system's output are used as input for future operations.[2] In AI, a feedback loop refers to the process of using the output of an AI system and corresponding user actions to train and reinforce models over time. The predictions and recommendations that are generated by AI are compared against the output, and feedback is provided to the model, making it learn from its errors. Feedback loops help AI systems learn what they did right or wrong, feeding them data that enables them to adjust their parameters to perform better in the future. This is a form of positive feedback loop that is sustained and supported by trust between humans and algorithms. A positive feedback loop is normally considered to have those components of a system that jointly increase each other's values when a stimulus occurs in one component. User profiles and recommendations shape a feedback loop. The users and the system are in a process of reciprocal confluence, where user profiles are continuously maintained and updated over time through interaction with the algorithm systems, and the quality of the algorithmic system is also improved by the user profiles.

Most AI applications in real-world systems follow this loop: data collection, data analysis, annotation and labeling, modeling, training, deployment, operationalization, sending the feedback into the system, and looping all over again. At every single point of this loop, bias can be added. Any model can spread biases in the training data. Humans should define their data sources and populations, their sampling methods for data collection, and the rules in the annotation and labeling task. The training model can proliferate and amplify the biases arising from prior analyses. The training model can enhance the feedback loop, which can contribute to bias. A relevant example of feedback loops producing bias can be found in the Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) study.[11] It predicts the probability of an inmate recommitting a crime in the future. By analyzing the COMPAS algorithms, it was revealed that the algorithms are not accurate and are biased toward colored races. Nonwhite subjects were falsely labeled as having a high risk of committing a crime. This can be inferred by the fact that the recidivism model can produce different results with different parameters, such as gender, location, probability of family members of convicts, and crime history. With these parameters, the models are biased toward specific ethnicities and races.

Generally, certain popular items are suggested selectively, while most other items are overlooked. These recommendations are then perceived by the users, and their reactions are logged and written into the system. This is called a negative feedback loop, which leads to generating and amplifying bias and misinformation. With negative feedback loops in

place, bias causes further bias. Once a bias is included in the system, it is not only compounded to produce discriminatory results that generate and amplify existing biases but also lead to algorithms developing their own ways to form biases. Because of the autonomous feature of negative feedback loops, algorithms can deviate from their intended goals, which can create significant obstacles when this bias mechanism goes unnoticed. Thus, designers and providers of algorithms should constantly monitor potential negative feedback loops that can lead to algorithms that generate biased decisions or even perpetuate gender and racial stereotypes.

FAKE NEWS, MISINFORMATION, AND AI

Fake news has proliferated rapidly online on social media platforms, partly because they can easily and quickly be shared with the advancement of algorithms[10]. "Fake news" exists within a larger ecosystem of mis/disinformation by having different taxonomy: false information, misreporting, polarized content, satire, persuasive information, biased commentary, and citizen journalism.[12] Fake news, misinformation, and recently, deep fakes have been a persistent concern ever since the advent of AI. Fake news has already fanned blazes of distrust toward the media, politics, and established ideologies around the world.[13] Despite heightened awareness and public concerns, the problem persistently continues, and no single method appears to work to dissipate the problem. In fact, it has increased with the advancement of AI technologies. Misinformation poses a considerable challenge to public debate during political campaigns, and disinformation about Ukraine's War with Russia has led to confusion and anxiety. For the last several years, the volume and dissemination of fake news and misinformation have grown considerably.[7]

Constant exposure to the same misinformation makes it likely that someone will accept it. If AI detects misinformation and cuts the rate of its circulation, this can stop the cycle of reinforced information consumption and dissemination patterns. ML algorithms can identify misinformation based on text nuance and the way stories are shared and distributed. However, AI detection on misinformation remains technically and operationally inaccurate. The contemporary algorithmic detection method is based on the evaluation of textual content. Although the method can determine the origin of the sources and the dissemination mechanism of misinformation, the essential limitation lies in how algorithms substantiate the actual nature of the information. Fake news and misinformation are more about philosophical questions of how people deal with the truth than technical or mathematical questions of algorithms and AI. Many organizations now have fact-checker software tools for identifying fake news for reporters and journalists. Most fact-checking software performs basic functions, such as content-independent detection, by using tools that target the form of the content and by using deep fake identifying tools to check any manipulated content, image, and video. However, algorithms cannot check whether the content itself provides false information. This task should be done by human fact-checkers by searching for social media posts or online information with similar queries and information.

Most current fact-checking approaches examine content by analyzing the semantic features of fake news. This approach may work at a basic level but faces bigger problems; for example, platforms like WeChat have language barriers, and fact-checkers cannot access the content of WhatsApp because it is an encrypted message. The reality is that detecting such news demands prior social, cultural, and political knowledge, which AI algorithms still lack. We can prevent someone from making fake news and from spreading and promoting misinformation by applying AI-powered analytics that uses anomaly detection, but eventually, AI cannot catch all problems related to fake news. Most current fake news detection systems require humans to work with AI to check the accuracy of information.

FAIRNESS AND TRANSPRENCY IN ALGORITHMS

Fairness and transparency are becoming important considerations in the use of algorithms for the recommendation and delivery of digital content.[14] Automated data gathering and sharing may involve processes that are unfair, flawed, opaque, or unaccountable. Over-the-top platform content recommendations embody these issues in highly visible applications. Fairness and transparency bring up

vital prerequisites in the design and development of algorithm-supported platforms, which are purportedly designed to offer accurate and reliable recommendations for users. It remains unresolved whether such recommendations match user interest, how the analytic processes are done, and whether the outcomes are legally responsible. Thus, fairness and transparency emerge as fundamental requirements in the use of algorithms on media platforms. When transparent, open, and fair services are provided, users are more likely to consider that the recommendations are of high quality. Highly transparent platforms can grant users a sense of personalization, as responsible and fair recommendations afford users a sense of trust that promotes satisfaction and a willingness to continue using them and subscribe to them. Open visibility and clear transparency of relevant recommendations boost the user interpretability of the system and search performance.[4]

AI incessantly affects the everyday lives of billions of media users. Algorithms are widespread and accepted in practice, but their popularity comes at the expense of limited transparency, systematic prejudice, and nebulous responsibility. Algorithmic filtering procedures may lead to more impartial, and thus possibly fairer, processes than those processed by humans. Yet, algorithmic recommendation processes have been criticized for their bias to intensify/reproduce prejudice, information asymmetry, distortion of facts, and the black-box process of decision-making. Algorithmic bias may increase algorithmic inequality in that ML automates and propagates unjust and discriminatory patterns.

REFERENCES

1. R. Benjamins, "A choices framework for the responsible use of AI," *AI Ethics*, vol. 1, no. 1, pp. 49–53, Feb. 2021, doi: 10.1007/s43681-020-00012-5.
2. D. Shin, B. Zaid, F. Biocca, and A. Rasul, "In platforms we trust? Unlocking the black-box of news algorithms through interpretable AI," *J. Broadcast. Electron. Media*, vol. 66, no. 2, pp. 235–256, Apr. 2022, doi: 10.1080/08838151.2022.2057984.
3. S. Herbet, *Models of Man*. New York, NY, USA: Wiley, 1957.
4. D. Shin, "How do people judge the credibility of algorithmic sources?" *AI Soc.*, vol. 37, pp. 81–96, Mar. 2022.
5. S. Sundar, J. Kim, M. Beth-Oliver, and M. Molina, "Online privacy heuristics that predict information disclosure," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, Apr. 2020, pp. 1–12, doi: 10.1145/3313831.3376854.
6. J. Zarocostas, "How to fight an infodemic," *Lancet*, vol. 395, no. 10225, p. 676, Feb. 2020, doi: 10.1016/S0140-6736(20)30461-X.
7. F. Huszar et al., "Algorithmic amplification of politics on Twitter," *Proc. Nat. Acad. Sci.*, vol. 119, no. 1, Dec. 2021, Art. no. e2025334119, doi: 10.1073/pnas.2025334119.
8. L. Sartori and A. Theodorou, "A sociotechnical perspective for the future of AI: Narratives, inequalities, and human control," *Ethics Inf. Technol.*, vol. 24, no. 1, pp. 1–11, Jan. 2022, doi: 10.1007/s10676-022-09624-3.
9. J. Dressel and H. Farid, "The accuracy, fairness, and limits of predicting recidivism," *Sci. Adv.*, vol. 4, no. 1, Jan. 2018, Art. no. eaao5580, doi: 10.1126/sciadv.aao5580.
10. N. Ahmad, N. Milic, and M. Ibahrine, "Data and disinformation," *Computer*, vol. 54, no. 7, pp. 105–110, Jul. 2021, doi: 10.1109/MC.2021.3074261.
11. M. Molina, S. Sundar, T. Le, and D. Lee, "'Fake News' is not simply false information: A concept explication and taxonomy of online content," *Amer. Behav. Scientist*, vol. 65, no. 2, pp. 180–212, 2021, doi: 10.1177/0002764219878224.
12. D. Shin, *Algorithms, Humans, and Interactions: How Do Algorithms Interact with People?* New York, NY, USA: Routledge, 2023.
13. S. Lewandowsky, U. K. Ecker, and J. Cook, "Beyond misinformation: Understanding and coping with the 'Post-Truth' era," *J. Appl. Res. Memory Cognit.*, vol. 6, no. 4, pp. 353–369, Dec. 2017, doi: 10.1016/j.jarmac.2017.07.008.
14. D. Shin and Y. Park, "Role of fairness, accountability, and transparency in algorithmic affordance," *Comput. Hum. Behav.*, vol. 98, pp. 277–284, Apr. 2019, doi: 10.1016/j.chb.2019.04.019.

DONGHEE SHIN is a professor at the College of Communication and Media Sciences, Zayed University, Dubai 144534, United Arab Emirates. Contact him at donghee.shin@zu.ac.ae.

EMILY Y. SHIN is a researcher at the Human–Computer Interaction Lab, Zayed University, Dubai 144534, United Arab Emirates. Contact her at emilyshin@acs.sch.ae.

IEEE COMPUTER SOCIETY

www.computer.org

PURPOSE: Engaging professionals from all areas of computing, the IEEE Computer Society sets the standard for education and engagement that fuels global technological advancement. Through conferences, publications, and programs, IEEE CS empowers, guides, and shapes the future of its members, and the greater industry, enabling new opportunities to better serve our world.

OMBUDSMAN: Contact ombudsman@computer.org.

CHAPTERS: Regular and student chapters worldwide provide the opportunity to interact with colleagues, hear technical experts, and serve the local professional community.

PUBLICATIONS AND ACTIVITIES

***Computer*:** The flagship publication of the IEEE Computer Society, *Computer*, publishes peer-reviewed technical content that covers all aspects of computer science, computer engineering, technology, and applications.

Periodicals: The IEEE CS publishes 12 magazines, 18 journals

Conference Proceedings & Books: Conference Publishing Services publishes more than 275 titles every year.

Standards Working Groups: More than 150 groups produce IEEE standards used throughout the world.

Technical Communities: TCs provide professional interaction in more than 30 technical areas and directly influence computer engineering conferences and publications.

Conferences/Education: The IEEE CS holds more than 215 conferences each year and sponsors many educational activities, including computing science accreditation.

Certifications: The IEEE CS offers three software developer credentials.

AVAILABLE INFORMATION

To check membership status, report an address change, or obtain information, contact help@computer.org.

IEEE COMPUTER SOCIETY OFFICES

WASHINGTON, D.C.:
2001 L St., Ste. 700,
Washington, D.C. 20036-4928
Phone: +1 202 371 0101
Fax: +1 202 728 9614
Email: help@computer.org

LOS ALAMITOS:
10662 Los Vaqueros Cir.,
Los Alamitos, CA 90720
Phone: +1 714 821 8380
Email: help@computer.org

IEEE CS EXECUTIVE STAFF

Executive Director: Melissa Russell
Director, Governance & Associate Executive Director: Anne Marie Kelly
Director, Conference Operations: Silvia Ceballos
Director, Information Technology & Services: Sumit Kacker
Director, Marketing & Sales: Michelle Tubb
Director, Membership Development: Eric Berkowitz
Director, Periodicals & Special Projects: Robin Baldwin

IEEE CS EXECUTIVE COMMITTEE

President: Hironori Washizaki
President-Elect: Grace A. Lewis
Past President: Jyotika Athavale
Vice President: Nils Aschenbruck
Secretary: Yoshiko Yasuda
Treasurer: Darren Galpin
VP, Member & Geographic Activities: Andrew Seely
VP, Professional & Educational Activities: Cyril Onwubiko
VP, Publications: Charles (Chuck) Hansen
VP, Standards Activities: Edward Au
VP, Technical & Conference Activities: Terry Benzel
2025–2026 IEEE Division VIII Director: Cecilia Metra
2024–2025 IEEE Division V Director: Christina M. Schober
2025 IEEE Division V Director-Elect: Leila De Floriani

IEEE CS BOARD OF GOVERNORS

Term Expiring 2025:
İlkay Altintaş, Mike Hinchey, Joaquim Jorge, Rick Kazman, Carolyn McGregor, Andrew Seely

Term Expiring 2026:
Megha Ben, Terry Benzel, Mrinal Karvir, Andreas Reinhardt, Deborah Silver, Yoshiko Yasuda

Term Expiring 2027:
Sven Dickinson, Alfredo Goldman, Daniel S. Katz, Yuhong Liu, Ladan Tahvildari, Damla Turgut

IEEE EXECUTIVE STAFF

Executive Director and COO: Sophia Muirhead
General Counsel and Chief Compliance Officer: Ahsaki Benion
Chief Human Resources Officer: Cheri N. Collins Wideman
Managing Director, IEEE-USA: Russell Harrison
Chief Marketing Officer: Karen L. Hawkins
Managing Director, Publications: Steven Heffner
Staff Executive, Corporate Activities: Donna Hourican
Managing Director, Member and Geographic Activities: Cecelia Jankowski
Chief of Staff to the Executive Director: Kelly Lorne
Managing Director, Educational Activities: Jamie Moesch
IEEE Standards Association Managing Director: Alpesh Shah
Chief Financial Officer: Thomas Siegert
Chief Information Digital Officer: Jeff Strohschein
Managing Director, Conferences, Events, and Experiences: Marie Hunter
Managing Director, Technical Activities: Mojdeh Bahar

IEEE OFFICERS

President & CEO: Kathleen A. Kramer
President-Elect: Mary Ellen Randall
Past President: Thomas M. Coughlin
Director & Secretary: Forrest D. Wright
Director & Treasurer: Gerardo Barbosa
Director & VP, Publication Services & Products: W. Clem Karl
Director & VP, Educational Activities: Timothy P. Kurzweg
Director & VP, Membership and Geographic Activities: Antonio Luque
Director & President, Standards Association: Gary R. Hoffman
Director & VP, Technical Activities: Dalma Novak
Director & President, IEEE-USA: Timothy T. Lee

IEEE

revised 14 January 2025

This article originally appeared in
Computer
vol. 57, no. 1, 2024

Generative Artificial Intelligence and the Future of Software Testing

Lucas Layman and Ron Vetter, *University of North Carolina Wilmington*

This virtual roundtable focuses on applications of generative artificial intelligence (GenAI) to software testing with four leading experts from the field. Our experts reflect on transforming the work of software testing with GenAI, its impact on quality assurance engineers, and privacy concerns.

Generative artificial intelligence (GenAI) is a branch of AI focused on models that create new content. GenAI models have been used to generate text from prompts, create images, formulate new molecules, and write program source code.[1] As GenAI's capability grows, it must address computational challenges, the scale of training data, and new aspects of trust, compliance, privacy, and ethics.[2,3]

The software engineering field is ripe for GenAI applications, including authoring specifications, generating test data, and writing program source.[4] Products such as GitHub Copilot[5] and Meta's CodeCompose[6] have already entered developers' tookits.

IMPACT ON SOFTWARE ENGINEERING

Computer: *In what ways do you anticipate GenAI will change how we engineer software?*

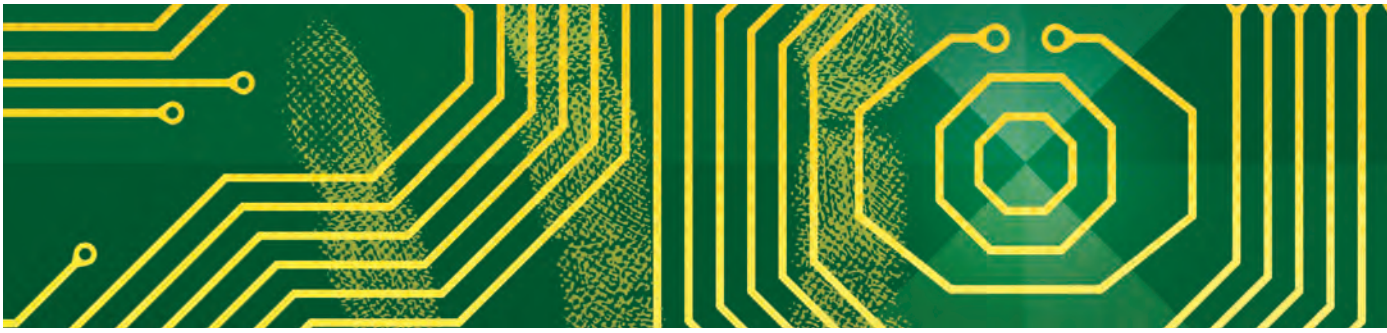
Thomas Dohmke: It will make programming more fun, allow engineers to be more ambitious, and make participating in software development—including testing of course—accessible to more people. Everyone who wants to should have the opportunity to be a developer. Given that tools like ChatGPT and GitHub Copilot allow us to interact with them in human language, in almost any human language, will allow more students to learn how to write software earlier in their lives. As such, AI will democratize access to software development and will significantly increase the number of people that have the skills to accelerate human progress.

Paul Gerrard: Improvements in the logical analysis of text will enable more effective critical evaluation of textual requirements, whether written in natural language or domain-specific languages like Gherkin. They will use examples to illustrate feature gaps, ambiguities, conflicts, and missing behaviors. These tools know more about business and application domains as those models will emerge over time. Security, reliability, availability, and failover testing will be supported or even performed by AI-based tools using model-based approaches. AI may also offer trustworthy guidance to stakeholders on the documentation, prioritization, and even repair of defects, and potentially the release-readiness of whole systems.

Adam Porter: GenAIs and other AI technologies are finding a wide range of applications in software engineering, just as they are in many other industries. In fact, we have already seen impressive applications of AI in code generation. More improvements are certainly on their way. I am particularly interested to see how GenAIs might be applied in other parts of the engineering process. Some candidates include performing business intelligence/gathering requirements for consumer applications based on mining open source data, creating better support for finding and configuring

Digital Object Identifier 10.1109/MC.2023.3306998
Date of current version: 5 January 2024

 Published by the IEEE Computer Society



ROUNDTABLE PANELISTS

Thomas Dohmke has been fascinated by software development since his childhood in Germany and has built a career building tools developers love and accelerating innovations that are changing software development. Currently, Thomas is chief executive officer of GitHub, where he has overseen the launch of the world's first at-scale artificial intelligence developer tool, GitHub Copilot, and now GitHub Copilot X. Before his time at GitHub, Thomas cofounded HockeyApp and led the company as CEO through its acquisition by Microsoft in 2014. He holds a Ph.D. in mechanical engineering from the University of Glasgow, U.K.

Paul Gerrard earned Masters degrees from the universities of Oxford and Imperial College, London. He has worked in software development and testing since the early 1980s as a developer and project manager and, for 30 years, a leading test consultant. He has chaired several international conferences, won three international awards, and founded the Test Management Forum, Technology Knowledge Base, and the Test Engineering Society. He has worked in projects of all sizes and criticality. Author and coauthor of several books, he focuses on professionalism and artificial intelligence in testing, and improving his poetry, drawing ability, and golf swing.

Adam Porter is a professor of computer science at the University of Maryland and the University of Maryland Institute for Advanced Studies. He holds appointments in the University of Maryland Institute for Systems Engineering and the University of Maryland Applied Research Laboratory for Intelligence and Security. He also serves as the executive and scientific director of the Fraunhofer USA Center MidAtlantic, a University of Maryland-affiliated applied research and technology transfer center specializing in software and systems engineering.

James Walker holds a Ph.D. in data visualization and machine learning in the field of visual analytics, a topic that combines human problem solving skills with the vast processing power of computers. James has given talks worldwide on the application of visual analytics and has several articles in high-impact journals. He has since applied these approaches to quality assurance, focusing particularly on model-based testing and test data management. He is the cofounder of Curiosity Software, a fast-growing start-up that helps enterprises drive quality throughout their software development lifecycle.

reusable software for specific use cases, and creating more effective software development education and team onboarding.

James Walker: Top engineers are integrating GenAI daily to accelerate writing code. As these AI models enlarge, and are trained on larger, richer datasets, their problem-solving capabilities and knowledge will grow. Numerous use cases exist for engineering tasks to automate code reviews, enhance code, query databases, and more. The understated problem of requirements quality often leads to unsuitable solutions and technical debt. These two areas present significant opportunities for AI: refining the formulation of complete requirements and addressing the problem of technical debt and understanding at a fine level through domain-specific large language models.

OPPORTUNITIES FOR TESTING

Computer: *What are the most exciting opportunities GenAI created for software testing? Can GenAI accelerate testing activity and improve test quality?*

Gerrard: The short answer is "yes, but…" I have used ChatGPT to scan HTML code to identify form fields, test data, and boundary values, create covering test cases and Python code to automate tests of simple transactions. But there are limitations in accuracy and comprehensiveness in such test design. Random/statistically based outputs mean responses are inconsistent, and the tool can forget what it has previously reported earlier in the same conversation. The tool can generate "ideas" for tests but needs careful prompting and supervision to check that it does not stray from the mission. It is almost human in its frailties.

Walker: The immediate opportunity is as an accelerator for quality, assisting with writing tests and code. The assets produced are not perfect, but they provide a great starting point. The longer-term opportunity lies in addressing technical debt. In large enterprises, the biggest challenge is understanding legacy systems/processes; there are pockets of knowledge, but they are siloed between teams and subject matter experts. Training AI in an organization can assist with understanding the landscape, allowing it to be tested appropriately. This is immensely empowering: AI would effectively become the hub of knowledge for driving understanding and promoting quality in an organization.

Porter: In its current state, GenAI seems to be very strong at conversation, summarization, and transformation (among many other things). Therefore, I expect that the initial applications of GenAI to software testing may revolve around these capabilities. For example, GenAIs could support conversational end user feedback and troubleshooting, providing highly contextualized data to the developers of a given software system. GenAIs can summarize large quantities of heterogeneous data, such as that found in software repositories, user and team Q&A forums, YouTube videos, requirements documents, and more. Finally, GenAIs transform information in one format to another, such as transforming usage scenarios and requirements statements into test artifacts and test code, generating test code for multiple different end user personas and goals, and translating test assets across different testing frameworks and toolsets.

Dohmke: GitHub Copilot has learned testing conventions from public code and various other texts in the model training set, such as blog posts, wiki pages, and documentation. It also has your project as added context. Whether you write a unit test first or the method, GitHub Copilot can use it to suggest the code for the respective other side. And this is just the beginning. With the help of GitHub Copilot, developers can generate many tests at the same time, and we will soon see the automatic generation of whole test suites.

PRIVACY AND CONFIDENTIALITY

Computer: *How will privacy and confidentiality concerns change when GenAI services are integrated into software testing?*

Walker: Organizations' back-end systems contain business rules, trade secrets, and the fundamentals of how an organization operates. Privacy and confidentiality should be of the greatest concern when they are trained and exposed to software testing data. Security risks include aiding hacking and potential exposure of trade secrets if models leak. Furthermore, unlawful use of sensitive information, for example, personally identifiable information, within applications is a concern. Legal and regulatory extensions, like the General Data Protection Regulation, need to be extended to cover AI use. Potential technical solutions may include on-premises [large foundation models] or sandboxed smaller models, and options for nonweight adjusting queries, safeguarding against breaches.

Gerrard: The training data that AI requires to deliver meaningful, reliable services to testers would need to include much proprietary data (code, usage patterns, architectural models, defect histories, etc.) collected across many organizations and systems. It's unlikely this will happen of course. It may be possible that some products appear trainable and usable within single organizations. But it seems unlikely a global "AI test model" could be created. Organizations sensitive to exposing their intellectual property and commercial activity to the outside world, will probably insist tools and models are for internal use only, within their own cloud infrastructure.

Dohmke: Ensuring user privacy and protecting user data are critical with the GenAI services in the market today, and it will remain critical when these services are integrated into software testing. Developers should take the time to understand how data flow through the GenAI services they use and make sure it fits their privacy needs. For example, with GitHub Copilot we never retain prompt data or suggestions for business users, and individual users must explicitly opt-in for us to retain prompt data. And, as GitHub is part of Microsoft, we adhere to the strict guidelines of the Microsoft Trust Code.

Porter: Privacy and confidentiality are critical concerns for this technology. Multiple public articles have shown cases in which GenAI users have effectively given their private information to the GenAI provider. This information was then used by the GenAI provider in ways that essentially made it public. One likely response will be that users create and manage their own private GenAIs, rather than rely on public providers. Interestingly, the open source community around GenAIs is flourishing and quite successful, lessening the need to interact with large GenAI providers.

BARRIERS TO ADOPTION

Computer: *What are the current barriers to GenAI adoption for software testing? What is required to address these challenges?*

Porter: As with many trendy technologies (e.g., Blockchain is one recent example) there's a real lack of understanding about what GenAI is, how it might actually be used, its benefits over existing technologies, and its potential downsides. This leads to magical thinking about potential use cases and applications in which GenAI can solve every problem that exists. There will need to be a careful examination of our software testing needs and processes, a thorough identification of GenAI strengths and weaknesses, a widespread exploration of specific use cases, and a data-driven comparison against existing solutions. We are only in the beginning stages of GenAI use. Much more experience and hard data will be needed before GenAI adoption becomes widespread.

Dohmke: Brains and GPUs. It'll take creativity to integrate GenAI into testing workflows and to build new AI-powered testing applications. It will also require calm consideration of risk and reward from companies and policymakers to not artificially block adoption. And of course, the world needs more GPUs to simply meet demand from software testing and every other field.

Gerrard: For too long, tool vendors have focused on the logistics of testing: test case management, test execution, defect reporting and management, and so on. With AI, vendors see low-hanging opportunities to, for example, make it easier to generate test automation code or test data. Help with such logistics is useful of course, but this does not help with the intellectual challenge of building test models from varying sources of knowledge, defining coverage measures, balancing test utility, coverage, and cost. We need to understand how testers think to identify requirements for true AI-based test assistants.

Walker: GenAI has the potential to hinder the testing industry. Testing aims to provide confidence to stakeholders that software functions correctly and adheres to requirements. AI is a black-box algorithm, harvesting inputs and providing outputs. Applying this to testing provides less transparency into the testing process, a lower understanding of methodologies applied, and no way to assess the quality of the test cases used (e.g., their coverage). The greatest barrier is comprehending the reasoning behind results and visualizing the generated data for user evaluation. Feedback loops, allowing users to input their subject-matter expertise and understanding to guide solutions, are crucial.

HOW WILL QUALITY ASSURANCE SKILLS CHANGE

Computer: *How will the skills required of software testers and quality assurance (QA) engineers change as GenAI tools integrate into the software engineering process? Will software testers and QA engineers become nonexistent?*

Gerrard: With the right tools, the skills profile of testers will change. They will become more valuable to software teams but that will mean fewer testers. The

best testers will develop a collaborative relationship with their AI partner. Testers will shift left to build relationships with stakeholders to refine system requirements and stakeholder needs for information from testing. New tools will capture models and data across the technical stack, the test team, test outcomes, for all time. The tools will develop both exploratory and advisory capabilities. They will make recommendations and with permission, run tests autonomously when they see opportunities.

I ENVISION THAT TESTING AND QA WILL BECOME MORE FOCUSED ON THE END-USER EXPERIENCE AND LESS FOCUSED ON CODE-CENTRIC ACTIVITIES, SUCH AS WRITING UNIT TESTS, AS GENAI TECHNOLOGIES CONTINUE TO MATURE.

Porter: GenAIs are just one of many technologies that have an impact on software testers and QA engineers (and nearly every other work category as well). Over time, we have repeatedly seen technology automating cognitively lower-level tasks, which pushes testers and QA engineers to focus on cognitively higher-level tasks. Software testers and QA engineers are not going away any time soon. I envision that testing and QA will become more focused on the end-user experience and less focused on code-centric activities, such as writing unit tests, as GenAI technologies continue to mature.

Walker: Testers reaping the benefits of GenAI have mastered effective prompt design. As AI integrates into testing tools, the barrier for leveraging AI will lower. However, the early adopters will hold a dominant position. I believe there will always be a place for QA engineers. QA engineers will always have a role in assuring stakeholders, fostering confidence, and applying critical thinking. Automation/testing/AI is a mechanism to provide confidence and answers. QA teams might diminish; however, there will always need to be owners of quality who make sure quality is addressed using the appropriate means.

Dohmke: GenAI makes software more useful, so it will increase demand for software and in turn drive demand for the people who help build it. The fundamental nature of roles and skills will change as we move toward testing GenAI-powered applications. Nearly everyone involved in software testing will be using GenAI in some form. Being skilled at prompting and understanding the output of the [machine learning] model or AI assistant will move the different roles closer together.

ETHICAL CONSIDERATIONS

Computer: *What are the ethical considerations that need to be addressed when deploying GenAI for software testing? How can organizations ensure fairness, transparency, and accountability in the testing process?*

Porter: GenAIs lack meaningful theoretical or empirical "guarantees" of many essential system properties, such as correctness, safety, fairness, high performance, and more. While their output is seductively human-like, no technology professional should be comfortable completely turning over critical functions to GenAIs. Without such guarantees in GenAIs themselves, additional safeguards will need to be built into GenAI applications. In some cases, these will be implemented as automated checks, safety shutoffs, manual reviews, and other approaches.

Walker: The internal workings of GenAI are opaque to the user, obscuring the decision-making process (i.e., black box). Reasoning and transparency are crucial for understanding why a specific output is given from a prompt, which can then be used to ensure fairness and accountability. As AI progresses, I anticipate a growing emphasis on visualization to help communicate these algorithms' inner workings. This improved transparency could subsequently allow us to better comprehend aspects of fairness and foster a sense of accountability within these systems.

Gerrard: Setting aside the obvious challenges of using, for example, production or personal data for testing, AI may have a role in protecting sensitive data. The bigger effect will be how we measure and improve the effectiveness of testers, our developers, and our processes.

The product of testing is information and only testing captures evidence of achievement. If "integrated systems intelligence" becomes available, AI can evaluate the performance of the test process against stakeholder needs for high-quality information. The value of testing is how insightful and actionable the product of testing—information—is to stakeholders.

Dohmke: Every use case is different, but broadly speaking I would encourage organizations to look to Microsoft's Responsible AI Standard[7] when deploying GenAI for software testing. It offers a clear path for methodically evaluating critical areas, like accountability, transparency, fairness, reliability, and safety.

FIVE–TEN YEAR OUTLOOK

Computer: *How do you see GenAI integrating into software testing processes five years in the future? 10 years?*

Dohmke: That will depend on what developers build. I believe we'll see a wave of tools that will transform every aspect of software testing within five years, if not faster. It will help with writing test cases, generate test cases automatically while checking test coverage, and identify untested areas of the codebase. It will also determine which tests to run against the set of changes, for example in a pull request, to shorten the turnaround times of large test suites. Adoption still takes time, but demand for more robust software and competitive pressures will result in GenAI being nearly universally adopted more quickly than, for example, [continuous integration/continuous deployment]. And will require little to no migration effort.

Gerrard: A tester uses their knowledge and experience, communication, and analytical skills to model usage patterns, failure modes (risks), required and conventional behavior, and scenarios to demonstrate the software "works" to enable testing stakeholders to make better-informed decisions. AI tools for testers will require integrated training data from code, changes, the old system, real-world data, usage patterns, test, and defect histories. AI could become a trusted partner of testers who explore knowledge sources and direct AI to perform much of the legwork of testing. But these tools need "integrated systems intelligence" and a focus on the thought processes of testers.

Porter: Although GenAIs applications are impressive now and destined to improve rapidly, I think their near-term use in software testing will be limited and narrow in scope. In particular, GenAIs lack meaningful theoretical or empirical safety "guarantees." That said, in the longer term, GenAIs and future AI innovations will be used to automate more and more currently manual tasks. Most interestingly, I believe that during this transition period, GenAIs will enable technically knowledgeable people to review and curate GenAI output in ways that leapfrog their productivity over less technically knowledgeable people.

Walker: Generative AI's future lies in specific models trained on organizational data to facilitate intelligent algorithms. The biggest barrier to that is a lack of structured data, which largely doesn't exist in the software domain. Over the next five to 10 years, I anticipate a shift toward prioritizing the harvesting of AI-training data from across the development lifecycle. Despite a current focus on models/algorithms, data are fundamental. I predict models will be trained on assets from throughout the software development lifecycle, enabling a comprehensive organizational AI. This could drive autonomous testing and quality assessment.

Computer: Many thanks to our panelists, who all agree that GenAI will transform the software engineering profession. In the near term, software testing will benefit from AI-based test case specification, test code authoring, and test data generation. However, test engineers will remain the ultimate authority who assure that AI-assisted software testing results in a reliable, safe, secure, and functionally correct system. The potential of GenAI for software testing, as in other disciplines, will be realized once organizations identify and overcome the limits of this technology for improving, rather than replacing, the practices of engineering.

REFERENCES

1. M. Jovanovic and M. Campbell, "Generative artificial intelligence: Trends and prospects," *Computer*, vol. 55,

no. 10, pp. 107–112, Oct. 2022, doi: 10.1109/MC.2022.3192720.
2. H.-Y. Lin, "Large-scale artificial intelligence models," *Computer*, vol. 55, no. 5, pp. 76–80, May 2022, doi: 10.1109/MC.2022.3151419.
3. Z. Akata et al., "A research agenda for hybrid intelligence: Augmenting human intellect with collaborative, adaptive, responsible, and explainable artificial intelligence," *Computer*, vol. 53, no. 8, pp. 18–28, Aug. 2020, doi: 10.1109/MC.2020.2996587.
4. I. Ozkaya, "Application of large language models to software engineering tasks: Opportunities, risks, and implications," *IEEE Softw.*, vol. 40, no. 3, pp. 4–8, May 2023, doi: 10.1109/MS.2023.3248401.
5. "Your AI pair programmer." GitHub Copilot. Accessed: Jul. 13, 2023. [Online]. Available: https://github.com/features/copilot
6. V. Murali et al., "CodeCompose: A large-scale industrial deployment of AI-assisted code authoring," May 2023. [Online]. Available: http://arxiv.org/abs/2305.12050
7. "Microsoft responsible AI standard, v2." Microsoft. Accessed: Aug. 30, 2023. [Online]. Available: https://blogs. microsoft.com/wp-content/uploads/prod/sites/5/2022/06/Microsoft-Responsible-AI-Standard0-v2-General-Requirements-3.pdf

LUCAS LAYMAN is an assistant professor of computer science at the University of North Carolina Wilmington, Wilmington, NC 28403 USA. Contact him at laymanl@uncw.edu.

RON VETTER is a professor of computer science and founding dean of the College of Science and Engineering at the University of North Carolina Wilmington, Wilmington, NC 28403 USA. He is a Senior Member of IEEE. Contact him at vetterr@uncw.edu.

Get Published in the *IEEE Transactions on Privacy*

This fully open access journal is soliciting papers for review.

IEEE Transactions on Privacy serves as a rapid publication forum for groundbreaking articles in the realm of privacy and data protection. Submit a paper and benefit from publishing with the IEEE Computer Society! With over 5 million unique monthly visitors to the IEEE Xplore® and Computer Society digital libraries, your research can benefit from broad distribution to readers in your field.

Submit a Paper Today!

Visit computer.org/tp to learn more.

IEEE COMPUTER SOCIETY | IEEE

EDITORS: Fabio Massacci, fabio.massacci@ieee.org
Eric Bodden, eric.bodden@uni-paderborn.de
Antonino Sabetta, as@sabetta.com

This article originally appeared in
IEEE SECURITY & PRIVACY
vol. 21, no. 5, 2023

DEPARTMENT: BUILDING SECURITY IN

Why Is Static Application Security Testing Hard to Learn?

Padmanabhan Krishnan and Cristina Cifuentes, *Oracle Labs*
Li Li, *Beihang University*
Tegawendé F. Bissyandé and Jacques Klein, *University of Luxembourg*

In this article, we summarize our experience in combining program analysis with machine learning (ML) to develop a technique that can improve the development of specific program analyses. Our experience is negative. We describe the areas that need to be addressed if ML techniques are to be useful in the program analysis context. Most of the issues that we report are different from the ones that discuss the state of the art in the use of ML techniques to detect security vulnerabilities.

While issues such as relevant datasets and representation of program semantics are common, our focus is on enhancing vulnerability detection by combining static analysis and ML approaches.

Static application security testing (*SAST*) is a methodology that statically examines source code to find security flaws that make the application susceptible to attack. SAST is popular because it can detect security vulnerabilities already in the early stages of the software development lifecycle. The static analysis can be integrated into a continuous integration/continuous delivery pipeline, thus automating the checks during the build process. While this is effective for deploying existing analyses, the process of developing new analyses is still manual: whenever a new defect or vulnerability type needs to be supported, an expert in static analysis needs to extend the existing framework to detect the new vulnerability type. This can be laborious and time-consuming, as one has to check that the new analysis has the desired accuracy, that it does not introduce any regressions to other analyses already deployed in production, and that it does not adversely affect the performance of the deployed products. Ideally, one would want to automate the generation of such analyses to make them available faster.

ML has been applied to perform security analysis tasks that are currently performed using static analyzers.[15] In particular, ML techniques have been used to learn to solve SAST problems. Actually, ML techniques have already become popular in the context of mimicking specific program analyses, such as symbolic execution. Although the results in such domains are impressive, they are, unfortunately, not generalizable to automatically learn static checkers, especially for security analysis. Even deep learning techniques have focused on relatively simple defect types, such as incorrect operators or assignments. Based on our experience in using different ML techniques for vulnerability detection, here we describe our insights into why it is hard to learn to solve SAST problems.

LEARN TO SOLVE STATIC ANALYSIS TASKS: OUR INITIAL ATTEMPTS

In our past work, we have combined program analysis with ML, aiming to enable the ML technique to

Digital Object Identifier 10.1109/MSEC.2023.3287206
Date of current version: 11 September 2023

 Published by the IEEE Computer Society

learn from existing program analysis approaches to improve future program analysis. In our experience, any simple combination of the two techniques does not work. ML by itself merely echoes its result with additional errors. Indeed, it often requires a program analysis to run in the first place and recognizes the output of the analysis as being reliable.[3] If we do not involve program analysis, the ML-based classifier can only use the input source code or its direct intermediate representation, like opcodes, to represent the program's semantics, such as loading data to a variable or calling a function (both considered as a sequence of words) to train and subsequently predict vulnerable code. Unfortunately, the tokens of the source code, such as opcodes, expressions, and statements, are generic syntactic constructs that alone do not represent the semantics of vulnerable code when directly applied to ML approaches.[2] While there are existing works that represent the code sequence with more advanced data structures, such as trees or graphs, e.g., using abstract syntax trees or program dependence graphs,[13] these statically extracted representations are not sufficient also to capture semantics related to the program's runtime behavior, such as values of expressions and operations on the heap. Whether ML techniques and lightweight program analysis techniques can be combined to have a technique that is comparable to a custom program analysis technique is still an open question.

For such techniques to be useful in practice, they have to work on real-world codebases, not toy programs. Our experience in using ML-based approaches on codebases with several millions of lines of code, such as Open Solaris to detect vulnerabilities in the C code, was not successful.[3] Since there is no ground truth in such large codebases, it is nontrivial to compute an F-score to evaluate the actual capability of the ML approaches. In the experiments we ran, the ML technique generated 250 times the number of potential vulnerability reports compared to a static analysis tool. However, the generated reports were all false positives, providing no useful information to the developers. This evidence experimentally shows that it is indeed nontrivial to learn ML approaches to handle real-world vulnerability detections.

Another key distinguishing feature between program analysis techniques and ML-based techniques is explainability. Program analysis techniques typically generate an abstract trace, also known as a *potential witness path*, derived from data-flow analysis to explain why a value generated at a program point can have a detrimental effect on an operation at another program point. While explainable ML is an active area of research, its focus is on generating explanations of the characteristics of the model that resulted in the observed output. No ML is able to produce abstract traces of program behaviors. This is related to the fact that the ML models do not capture the execution semantics of programs (i.e., runtime behavior). There are ML techniques that accept traces as input, but none of them, as yet, can generate traces from programming language models.

WHETHER ML TECHNIQUES AND LIGHTWEIGHT PROGRAM ANALYSIS TECHNIQUES CAN BE COMBINED TO HAVE A TECHNIQUE THAT IS COMPARABLE TO A CUSTOM PROGRAM ANALYSIS TECHNIQUE IS STILL AN OPEN QUESTION.

UNDERSTANDING THE LIMITATIONS OF ML APPLIED TO STATIC ANALYSIS

In this section, we summarize, at a high level, the reasons the ML-based techniques fare poorly. These results are based on our experience of exploring such techniques in different domains, including misuse of cryptographic application programming interfaces (APIs) in Java programs and detecting memory-related issues in C programs.

Labeling Issue: Learning Through Crowd-Sourcing Solutions Is Not Feasible

The usual ML problem of having sufficient labeled data is a challenge. Creating such labeled data of a large corpus of code with annotated vulnerabilities can hardly be automated. It is unclear whether creating such labeled data is any cheaper than writing a specific program analysis. Solutions, such as

crowd-sourcing (e.g., learning from existing benchmarks collected by different teams from different code repositories), which works for very simple problems, do not typically work in the context of security analysis: in fact, it can be hard to get consensus on vulnerable code purely through such crowd-sourced datasets (e.g., such datasets per se may suffer from quality issues[14]). Our work on the misuse of cryptographic APIs[4] shows that the level of expertise required to identify proper and improper uses is quite high. While researchers have shown that it is possible to learn rules from code changes to fix incorrect cryptographic API usages, it is nevertheless hard to automate the process to achieve a comparable set of rules manually summarized by humans. By checking the updates of API usage in the evolution of 40,000 real-world Android apps, we have experimentally found that cryptographic APIs are widely misused in practice. Such misuses are not even regularly fixed by app developers.

This labeling issue also relates to a "definition issue," where several security-related concepts are not well-defined. In contrast with other traditional classification or detection tasks on which artificial intelligence techniques perform extremely well, several important security concepts are difficult to define properly, and they are often context-dependent. Researchers and analysts still require a lot of effort and expertise to check if a given warning is actually a malicious piece of code or a vulnerability.

Semantics Issue: Learning Code Semantics Is Hard

While pretrained models like CodeBERT[1] and Graph CodeBERT offer promising new code representations (i.e., new embeddings for code), they still only capture the structural aspects of the code. They do not quite capture the runtime semantics of the code, especially for arithmetic expressions and operations on arrays. Other approaches like JSNice[5] do guess the semantics, but that is only in the context of variable renaming; i.e., they do not deal with the semantics of the instructions in the code. Yamaguchi et al.[13] propose a novel code representation approach called *code property graph* that merges abstract syntax trees, control flow graphs, and program dependence graphs into a joint data structure, representing the semantics more comprehensively. Their approach, however, requires performing complicated program analysis already (e.g., to build data-flow analysis) and it is hard to retain context-sensitivity information (which has been considered important for purely static program analysis approaches).

Assessment Issue: In the Lab Versus in the Wild

It is not rare to read papers proposing a new ML-based approach to solve a given security problem, for instance, malware detection, showing impressive performance scores, sometimes up to 0.99.[8] We have shown in in Allix et al.[6] that most of these approaches suffer from assessment issues. Indeed, many approaches are assessed with what we call *in the lab* validation scenarios, i.e., a combination of 10-fold cross-validation and a limited dataset. We demonstrated the limitations of such a validation scenario. In particular, we showed that 10-fold cross-validation on the usual sizes of datasets presented in the literature is not a reliable performance indicator for realistic malware detectors "in the wild." With Tesseract, Pendlebury et al.[7] confirmed our findings and introduced the notions of *spatial bias* (distributions of training and testing data that are not representative of a real-world deployment) and *temporal bias* (incorrect time splits of training and testing sets). In the context of program semantics, focusing only on specific datasets like big data clone benchmarks,[10] semantic clone bench,[11] or using examples from GoogleCodeJam[12] seems to yield good results. But these articles do not investigate the case when a model is generated on one benchmark and is used on a different set of benchmarks. Thus, is it not possible to estimate the generalizability of the approaches.

Understanding the Dataset's Diversity Is Challenging

Another important aspect of the evaluation of learning-based techniques is the diversity of data within the dataset (e.g., to what extent has the dataset covered the landscape of the concerned problems). For instance, in the context of vulnerability detection, the commonly used dataset contains only code fragments that are related to vulnerabilities. Hence,

any evaluation that uses only that dataset is potentially misleading. It is important to use datasets that also have nonvulnerability-related code fragments. Furthermore, the number of nonvulnerability-related code fragments must be much higher than vulnerability-related code fragments. This will determine if the proposed technique is actually applicable in practice. That is, the technique must be able to distinguish vulnerability-related code from nonvulnerability-related code where most of the code is not vulnerable.

Lack of Explainability

Program analysis approaches usually yield warnings with relevant data-flow traces and even change recommendations that are often useful for users to understand the problem or fix the issues. This level of explainability as to why the program analysis determined that a particular statement in the code is an issue is not available when performing ML classifiers, they only report there is likely a vulnerability but do not explain why it is regarded as such. Therefore, we argue that, in order to make ML approaches more useful in practice, it is important to develop explainable ML techniques.

It is indeed hard to train ML-based security static checkers. We have identified four main reasons that make learning static security checkers challenging: labeling, semantics, assessment issues, and explainability. The labeling issue can be overcome by putting more effort into building reliable artifacts, sharing annotated datasets, releasing tools, etc. This is still too rarely done in the security community. The semantics issue can be addressed by developing new advanced code representation techniques, for instance, by embedding semantically rich information such as value-flow graphs. Regarding the assessment issue, we strongly invite researchers to adequately assess their approach to match practical and realistic constraints. Finally, the lack of explainability is a tough area of research where we invite researchers to develop techniques to generate traces from programming language models.

Moreover, while it is nontrivial to automatically learn to generate fully functional SAST approaches, we argue that it might still be feasible to generate partial solutions, e.g., only using ML to generate modules (i.e., type inference module of a static analysis approach) that are actually suitable for ML approaches. These partial modules could then be integrated into program analysis approaches to enable better performance, which cannot be achieved using program analysis techniques alone. Our fellow researchers have recently demonstrated the feasibility of implementing that.[9] They have proposed an approach that leverages deep learning techniques to infer types for Python programs and then integrates the outcomes into a program analysis approach to validate and refine the results. Static analyzers could further leverage this ML-generated type data to support more advanced program analyses, such as context-aware data-flow analysis. We invite the research community to further explore this exciting research direction.

REFERENCES

1. Z. Feng et al., "CodeBERT: A pretrained model for programming and natural languages," in *Proc. Findings Assoc. Comput. Linguistics, EMNLP*, 2020, pp. 1536–1547, doi: 10.18653/v1/2020.findings-emnlp.139.
2. T. Chappell, C. Cifuentes, P. Krishnan, and S. Geva, "Machine learning for finding bugs: An initial report," in *Proc. IEEE Workshop Mach. Learn. Techn. Softw. Qual. Eval. (MaLTeSQuE)*, 2017, pp. 21–26, doi: 10.1109/MALTESQUE.2017.7882012.
3. Y. Zhao, X. Du, P. Krishnan, and C. Cifuentes, "Buffer overflow detection for C programs is hard to learn," in *Proc. Companion (MLPL) ISSTA/ECOOP*, 2018, pp. 8–9, doi: 10.1145/3236454.3236455.
4. J. Gao, P. Kong, L. Li, T. F. Bissyandé, and J. Klein, "Negative results on mining crypto-API usage rules in android apps," in *Proc. IEEE/ACM 16th Int. Conf. Mining Softw. Repositories (MSR)*, 2019, pp. 388–398, doi: 10.1109/MSR.2019.00065.
5. V. Raychev, M. Vechev, and A. Krause, "Predicting program properties from 'Big Code'," *ACM SIGPLAN Notices*, vol. 50, no. 1, pp. 111–124, Jan. 2015, doi: 10.1145/2775051.2677009.
6. K. Allix, T. F. Bissyandé, Q. Jérome, J. Klein, R. State, and Y. Le Traon, "Empirical assessment of machine learning-based malware detectors for android: Measuring the gap between in-the-lab and in-the-wild validation scenarios," *Empirical Softw. Eng.*, vol. 21,

no. 1, pp. 183–211, Feb. 2016, doi: 10.1007/s10664-014-9352-6.
7. F. Pendlebury, F. Pierazzi, R. Jordaney, J. Kinder, and L. Cavallaro, "TESSERACT: Eliminating experimental bias in malware classification across space and time," in *Proc. 28th USENIX Security Symp. (USENIX Security)*, Santa Clara, CA, USA, 2019, pp. 729–746.
8. Y. Liu, C. Tantithamthavorn, L. Li, and Y. Liu, "Deep learning for android malware defenses: A systematic literature review," *ACM Comput. Surv.*, vol. 55, no. 8, pp. 1–36, Dec. 2022, doi: 10.1145/3544968.
9. Y. Peng et al., "Static inference meets deep learning: A hybrid type inference approach for Python," in *Proc. 44th Int. Conf. Softw. Eng. (ICSE)*, 2022, pp. 2019–2030, doi: 10.1145/3510003.3510038.
10. J. Svajlenko, I. Keivanloo, and C. Roy, "Big data clone detection using classical detectors: An exploratory study," *J. Softw., Evol. Process*, vol. 27, no. 6, pp. 430–464, Jun. 2015, doi: 10.1002/smr.1662.
11. F. Al-Omari, C. K. Roy, and T. Chen, "SemanticCloneBench: A semantic code clone benchmark using crowd-source knowledge," in *Proc. IEEE 14th Int. Workshop Softw. Clones (IWSC)*, 2020, pp. 57–63, doi: 10.1109/IWSC50091.2020.9047643.
12. W. Wang, G. Li, B. Ma, X. Xia, and Z. Jin, "Detecting code clones with graph neural network and flow-augmented abstract syntax tree," in *Proc. SANER*, 2020, pp. 261–271, doi: 10.1109/SANER48275.2020.9054857.
13. F. Yamaguchi, N. Golde, D. Arp, and K. Rieck, "Modeling and discovering vulnerabilities with code property graphs," in *Proc. IEEE Symp. Security Privacy*, 2014, pp. 590–604, doi: 10.1109/SP.2014.44.
14. Y. Zhao et al., "On the impact of sample duplication in machine-learning-based android malware detection," *ACM Trans. Softw. Eng. Methodology*, vol. 30, no. 3, pp. 1–38, May 2021, doi: 10.1145/3446905.
15. T. Marjanov, I. Pashchenko, and F. Massacci, "Machine learning for source code vulnerability detection: What works and what isn't there yet," *IEEE Security Privacy*, vol. 20, no. 5, pp. 60–76, Sep./Oct. 2022, doi: 10.1109/MSEC.2022.3176058.

PADMANABHAN KRISHNAN is the director of research at Oracle Labs, Brisbane, QLD 400, Australia, leading a team that is working on developing suitable tools to detect and remediate security vulnerabilities. His research interests include program analysis, application security, and formal methods. Krishnan received a B.Tech. from the Indian Institute of Technology of Kanpur, India and a Ph.D. in programming languages from the University of Michigan, USA. He is a senior member of the Association of Computing Machinery and a Senior Member of IEEE. Contact him at paddy.krishnan@oracle.com.

CRISTINA CIFUENTES is vice president of software assurance at Oracle Labs, Brisbane, QLD 400, Australia, leading a global team focusing on making application security and software assurance, at scale, a reality. Her research interests include software assurance. Cifuentes received a Ph.D. in computer science from the Queensland University of Technology. She was the founding Director of Oracle Labs, Australia in 2010 and has over 20 years of industrial experience, holds 15+ US patents, and has published over 50 peer-reviewed publications. Contact her at cristina.cifuentes@oracle.com.

LI LI is a professor with the School of Software, Beihang University, 100191 Beijing, China. His research interests include mobile software engineering and intelligent software engineering. Li received a Ph.D. in software engineering from the University of Luxembourg. He is a Senior Member of IEEE. Contact him at lilicoding@ieee.org.

TEGAWENDÉ F. BISSYANDÉ is a chief scientist, associate professor at the University of Luxembourg, L-1359 Luxembourg, Luxembourg, where he conducts research on program debugging and repair at the Interdisciplinary Centre for Security, Reliability, and Repair. His research interests include program repair and software analytics. Bissyandé received a Ph.D. in computer science from the University of Bordeaux, France. He is a Member of IEEE. Contact him at tegawende.bissyande@uni.lu.

JACQUES KLEIN is a full professor in software engineering and software security within the Interdisciplinary Centre for Security, Reliability, and Trust at the University of Luxembourg, L-1359 Luxembourg, Luxembourg. His main research interests are software security, software reliability, and data analytics. Klein received a Ph.D. in computer science from the University of Rennes, France. He is a Member of IEEE. Contact him at jacques.klein@uni.lu.

Unlock Your Potential

WORLD-CLASS CONFERENCES — Stay ahead of the curve by attending one of our 195+ globally recognized conferences.

DIGITAL LIBRARY — Easily access over 900k articles covering world-class peer-reviewed content in the IEEE Computer Society Digital Library.

CALLS FOR PAPERS — Discover opportunities to write and present your ground-breaking accomplishments.

EDUCATION — Strengthen your resume with the IEEE Computer Society Course Catalog and its range of offerings.

ADVANCE YOUR CAREER — Search the new positions posted in the IEEE Computer Society Jobs Board.

NETWORK — Make connections that count by participating in local Region, Section, and Chapter activities.

Explore membership today at the IEEE Computer Society

www.computer.org

IEEE COMPUTER SOCIETY

IEEE

EDITOR: Hallam Stevens, hallam.stevens@jcu.edu.au

DEPARTMENT: BIOGRAPHIES

This article originally appeared in
IEEE Annals of the History of Computing
vol. 45, no. 4, 2023

Monte Sala's Cryptographic Achievements

T. Alex Reid, University of Western Australia, Crawley, WA, 6009, Australia

This article sets out the principal achievements of an Italian-born Australian inventor, Monte Sala, who exhibited an ability to solve complex electronic problems and to design and build world-class devices. His career spanned radio and TV, NASA space tracking, vision research experiments, telemetry, Pay-TV, and cyrptography, all the time designing, developing, manufacturing, and promoting electronic devices of all kinds. He established a fledgling electronics industry in Western Australia (for which he was awarded the Order of Australia), and his career reached its climax with the invention of devices for encrypting data on communications lines—these were sold worldwide, despite some high-level opposition, culminating in adoption by SWIFT, the international bank clearinghouse.

MIGRATION TO AUSTRALIA

Amedeo ('Monte') Filiberto Sala (aka Amedeo Sala-Spini) was born in Trieste in 1927, and was a post-war émigré from Italy, arriving in Australia in 1950 [24], [25]. He had started formal civil engineering qualifications in Italy, but the onset of World War II prevented him completing them. Upon arriving in Australia, he was housed at the Bonegilla Migration Hostel, where he was taught some basic English. He then fulfilled his obligation as an assisted migrant by working at the Melbourne and Metropolitan Board of Works at its Werribee Sewage Works. He moved to Western Australian (WA) in 1952, hoping to find work in the mining industry with the assistance of relatives in Kalgoorlie. But he did not like that temporary work and moved to Perth, where opportunities in radio and television opened up.

During the period 1954 to 1961, Sala moved from radio technician into electronics, working with various audio equipment and later TV. He worked for several companies, and finally for Amalgamated Wireless Australasia (AWA). There he was given the task of installing the first computer in WA, a Bendix G15 acquired by the Main Roads Department in March 1962 [15], [24], gaining competence with this computer [7].

1058-6180 © 2023 IEEE
Digital Object Identifier 10.1109/MAHC.2023.3322950
Date of current version 23 November 2023.

NASA CARNARVON SPACE TRACKING STATION

AWA won a contract to provide personnel to NASA, and Sala was seconded in 1963 to work at the Carnarvon Tracking Station, 900 km North of Perth [5], [24], [25]. He received intensive and high-quality training for this role from NASA, first at Muchea in WA, and then in the USA, to become a Digital Command System Engineer. He worked at the NASA Carnarvon Tracking Station for 6 years in a senior engineering role during the unmanned and manned Gemini and Apollo missions that were heavily reliant on radar tracking. He rose to recognition within NASA circles for his ability to improve the performance of equipment.

One device he developed was a typical example of Sala's ability to adapt, design, and implement innovative solutions. He adapted superseded equipment from the decommissioned Mercury tracking station at Muchea, to develop a converter for radar track data into teletype signals, which could be transmitted instantaneously to the Goddard Space Flight Center in Maryland. NASA approved the design for use at the Wallops Island Virginia Tracking Station and elsewhere, and it was used by NASA across the unmanned Saturn SA-5 and SA-6 missions [12]. In 1969, Sala received a NASA Apollo Achievement Award for advancing "... *the nation's capabilities in aeronautics and space* ..."

 Published by the IEEE Computer Society

Monte Sala in 1966 at the Carnarvon Tracking Station Fountain he built: National Archives of Australia: A12111, 1/1966/16/102.

THE UNIVERSITY OF WESTERN AUSTRALIA: BETAGRAPH AND PICTURE TRANSMISSION SYSTEM

In 1969 Sala moved to Perth, mainly for the sake of his children's education, and secured a position as Computer Operations Manager at the Computing Centre, University of WA. There was limited opportunity within the Computing Centre for Sala to employ his inventive skills, as it mostly involved running large computers for researchers and students in a production environment. However, the University at that time had acquired a Digital Equipment Corporation PDP-6, with users connected over telephone lines to the computer [15]. Users remote from the computer found they could not get reliable connections. Sala quickly designed and built a number of modems, which enabled reliable connection to the computer over great distances.

His inventive skills quickly led him to a close association with the Psychology Department, which was looking to employ electronics in a range of experiments and laboratories. There he designed and built a number of electronic devices for use in their experiments, such as timing people's responses to visual stimulation. The head of Psychology at the time was Professor John Ross, who had a particular interest in visual perception, and with whom Sala struck up a working relationship and friendship [18]. By 1973, Sala left the Computing Centre and was given by the University a roving commission to pursue the sort of alliance he had with Psychology.

In the course of his research into visual perception, Ross discovered a remarkable capacity of the human visual system to "fill in" missing information, as when a moving object like a car is seen through a row of pickets [2]. Ross discovered that the eye's ability to construct a picture of the moving object is very highly developed indeed, interpolating what it does see, whether the image is comprised of text or pictures [17].

Ross and Sala conceived and built a device they called the Betagraph to exploit this capability. It was composed of columns of lights, which corresponded to the gaps in the picket fence. The eye sees a moving image, which fills the entire space in the display. It enabled information to be conveyed with a lot less input than normal, could be controlled by computer, and consumed much less power, providing a significant advantage over conventional displays of that time (1976). A wide range of valuable applications were envisaged for the Betagraph [20], [21], [32].

Based on this research into visual perception, Sala also devised a device for transmitting images over communication lines. The system developed by the Sala and Ross partnership broke the picture down into thousands of individual picture elements, which were transmitted one by one in a pseudorandom order. Each picture element carried a tag or label indicating its location in the picture. The receiver used the label to put the picture elements in their proper place within the picture as they arrived. In terms of hardware cost, speed, reliability, and security it was superior to anything previously available [19], [32].

DELTEC ELECTRONICS AND TELEMETRY

In 1972 Sala set up his own company, Deltec Pty Ltd, to undertake the manufacture and marketing of designs based on his inventions, including the modems described above. The Picture Transmission System and the Betagraph were both developed in association with Deltec; the University, as the intellectual property owner, lodged patent applications [19], [20], [21]. In February 1976, these two inventions were exhibited at the World Fair for Technology Exchange held in Chicago, where they attracted considerable attention and publicity [35].

From 1972, Sala combined his work at the University with supervising the engineering team at Deltec which undertook the development, manufacture, and marketing. Sala believed passionately in close links between academia and industry, and argued repeatedly that an electronics industry bringing together universities, industry, and the government to support innovative research and application would be viable in Western Australia [23]. By 1977, with the support of the University, he left the University to join Deltec on a full-time basis.

Marketing of the Betagraph and the Picture Transmission System stalled, as the University was unsure of how to progress this newly developed intellectual

Monte Sala outside the University of Western Australia in 1979 [Used by permission of Symbion Health Ltd, copyright owner].

property, though some sales were made [8]. Nevertheless, convinced that spin-offs from innovations often had a much better chance of commercial success than the original invention, Sala set out with an idea derived from the Picture Transmission System development. This was in telemetry.

This work started when Sala, on behalf of Deltec, responded to a 1977 tender from Mount Newman Mining Company who were concerned at the number of derailments on their long iron ore trains in WA's Pilbara region. This sort of telemetry system was Sala's forte, and he made a successful bid for the work. Deltec subsequently tendered for and achieved sales of telemetry systems and optical fiber communications to many government and industry players, including most Fire Brigades around Australia, Security Monitoring firms, the Department of Foreign Affairs at their offices in Washington, London and Canberra, and many other clients [22].

In addition to his telemetry systems, Sala also developed a Random Noise Generator, producing long sequences of truly random values derived from a thermal noise circuit via a semiconductor. Based on this, Sala designed, built, and sold automatic prize drawing systems for the Lotteries Commission in WA and in Queensland [30].

In 1978, the company Mayne Nickless Ltd, offered to buy into Deltec, acquiring a majority shareholding, and renaming it Deltec Pacific Pty Ltd. This provided the capital and marketing clout that Deltec needed.

With this backing, Deltec won contracts to the value of several million dollars. With a broad range of products on offer, Deltec was now enjoying considerable commercial success. However, the company pursued directions which did not necessarily align with those of Sala, and still with many other ideas awaiting his attention, he departed Deltec Pacific in early 1980, and set up another company, Ran Data, as a vehicle to pursue these ideas.

ENCRYPTION, RAN DATA, AND MERRILL LYNCH

In keeping with his aim of pursuing spin-off opportunities, Sala exploited another by-product of the work on the Picture Transmission System, which involved the random number generator he had developed. This was the burgeoning new field of Subscriber Pay TV, where TV transmissions needed to be secured to prevent nonsubscribers from gaining access, in a way that was both fast and inexpensive.

By mid-May 1980, Sala demonstrated to the Director of Government Computing for WA, Dennis Moore, a fully operational Subscriber Pay TV system incorporating an early form of his encryption algorithm [11]. Moore asked for an independent assessment, so Sala organized for Howard Shephard, a television consultant and former chief engineer and general manager for a commercial TV station in Perth, to inspect the system. Shephard stated that the quality of the reconstructed picture after descrambling (the term then used) was better than anything he had ever seen [29].

However, Moore felt that Sala should consider a more general application—the design of an encryption system to provide privacy of sensitive information on common carrier links, such as those used in the government's networks [11]. Sala accepted the challenge, saying, "I can do that!," and within a few days adapted the Subscriber Pay PV scrambler to develop and demonstrate two working preproduction prototypes. These units had the encryption and decryption embedded in hardware, employing the "one-time pad" principle. When tested on several computer systems operated within the government, they performed well and worked every time.

Soon after, in search of investment funds, Sala was introduced to Merrill Lynch Pierce Fenner and Smith (Merrill Lynch) in New York who suggested Sala visit them, which he did in June 1980. A series of intensive meetings with technical and intellectual property experts was arranged by Merrill Lynch. It became clear that the broad field of Subscriber Pay TV presented the most immediate opportunity with Sala describing his solution for a reliable, secure, and fast method of addressing subscribers.

Over the next two years, detailed discussions took place with Merrill Lynch, including the development by McKinsey & Co of a pro-forma business plan and marketing plan [9]. These considered three key market areas for communication network devices, being Cable Television (the new term for Subscriber Pay TV), telemetry, and general data communications encryption. Cable Television was considered to have high barriers to entry, so the business plan was adapted to focus on telemetry and data encryption. After completing its due diligence, Merrill Lynch took the rare step of investing in Ran Data itself [6]. The investment was $US1.2 million, which gave it a 13% minority shareholding.

With a business plan in place and investment capital secured, the marketing of Ran Data's encryption technology began at pace. Inevitably, of course, potential buyers needed to be reassured of the security of the encryption, which Sala called the Entropic Key Encryption System (EKES). It was implemented in hardware, and was an improvement on the original "scrambling" system, based on a one-time pad. Ran Data contracted the statistical and mathematical consultancy company Siromath Pty Ltd to undertake a comprehensive study, which stretched to mid-1984 and employed more than nine months of senior consultant Dr. Geoff Riley's time [16].

Ran Data made contact with Professor Adi Shamir, the internationally acclaimed cryptanalyst, and co-founder of RSA, one of the first public-key encryption systems, and engaged him to review Siromath's analysis, recognizing the value that having Shamir's imprimatur for EKES would bring. Sponsored by Ran Data, Shamir visited Perth in July 1983 to conclude his review of EKES and to deliver a series of lectures on data encryption and communications security [31].

At the time, the "gold standard" for encryption was the Data Encryption Standard (DES), established by the National Bureau of Standards in USA, and first published in 1977. It used a 56-bit key which was secret to the user and was always a topic of discussion given its short key length. In contrast, EKES had a 32,000-bit key, which was generally accepted as being uncrackable [28]. Ran Data incorporated the DES into its system as an optional extra.

Accordingly, in-depth analysis of the statistical and mathematical properties of EKES was undertaken by Siromath [16] and reviewed by Shamir [28]. The analysis included software simulations of discrete EKES components and various combinations of these components. Various mathematical analyses of the general structure of the EKES algorithm were undertaken, including randomness tests on the output sequences. It was also subjected to a variety of specialized cryptanalytic attacks. Shamir concluded that EKES did not have any statistical or mathematical weaknesses which could compromise its security, noting the robustness of the basic design and multiple protection layers. In particular, he observed the strength of having such a long key length, making EKES superior to DES. Integrating DES into the EKES algorithm combined the benefits of both.

Ran Data Encryptor Model 183 promotional brochure [13].

Shamir was convinced that the EKES algorithm was a "well-designed and thoroughly checked cryptosystem, which can safeguard highly sensitive information for long periods of time against sophisticated opponents" [28]. Sala was forever proud of that achievement. He did not patent this algorithm design, receiving advice from a leading New York patent attorney that the best form of intellectual product protection in this instance was to keep the design secret.

In addition to this research and analysis, Ran Data continued in various ways to cement its status as a world leader in the field. For example, it undertook to sponsor a Research Scholarship at Deakin University in Geelong, commencing in 1984, under the supervision of well-known computer security researcher, Professor B. J. Garner.[1]

This relationship benefited both parties and was in line with Sala's long-held belief in synergy between researchers and industry.

[1]Deakin Scholarship sponsored by Ran Data, established with the Foundation Chair of Computing, Professor B. J. Garner, for a 3-year period initially, with a focus on "high speed digital communications with particular reference to algorithm design and key management principles". (Correspondence dated 6 December 1988 held in Tony Sala's private collection).

Given the favorable reviews by such reputable authorities as Shamir, the Ran Data encryption technology began to make strategic inroads both in Australia and abroad, as significant numbers of orders started to build [13], [14].

EXPORT ROADBLOCKS

By late 1984, Ran Data had become a publicly listed company, and its customer base was a "who's who" of government and commerce in Australia. This covered some 50 high profile organizations that included police departments, royal commissions, customs offices, breweries, data communications organizations, news services, attorney-general departments, stock exchanges, taxation offices, large mining companies, credit unions and banks of all descriptions, including the Reserve Bank of Australia. The customer base also included a growing list of international organizations with similar needs for data security. The glowing testimonials, the high profile customer organizations, and the volume of orders inevitably attracted attention from the media, and from other more secretive agencies.

In 1983, if a company wanted to export encryption products, it was necessary to obtain export approval from the Australian Department of Defence Support. This was done by having end-user certificates approved by the Department, which required, in turn, the Minister for Defence's discretionary decision on a case by case basis. There were no guidelines at that time for the processing of end-user certificates for export licenses generally, and there were also no avenues of appeal if export licenses were refused. This was a new area both for the government and for business. It was quite a revelation to Ran Data that encryption equipment was considered to be "defense materiel" [4].

In February 1984, the Australian Defence Signals Directorate Intelligence Agency (DSD) made itself known to Ran Data. It said it needed to evaluate the then new Ran Data EKES-183 algorithm for government use. DSD had been involved with the evaluation of the EKES-181 devices that were in use with the Australian Federal Police. This first visit by the agency was a portent of things to come, as they talked about countries deemed "grey" to which exporting the Ran Data encryptors would not be permitted. In addition, to Ran Data's considerable surprise, before long there was talk of designing a special device for export markets.

DSD wanted the encryption algorithm to incorporate a modification which Ran Data believed might introduce a weakness into the system. They also proposed modifications to other system components. Ran Data had to balance not alienating DSD, which might result in the export market being reduced only to "profiled" users, with not compromising the integrity of its encryption. Ran Data was selling to multinational corporations, who had the resources to undertake in-depth analyses of these devices, and would inevitably discover any compromise that would likely have had a significant impact on the company's credibility and sales.

Indeed, already approvals to "unprofiled" users were being caught up in a convoluted bureaucratic process. Additionally, similar intelligence agencies in the U.K. had by now also become involved, further complicating the sale of equipment from Ran Data's U.K. office, which was responsible for the European market.

In the end, Ran Data resolved not to make any of the proposed amendments to its design. This meant that it was now only able to export to prequalified financial institutions and other "profiled" users. This considerably narrowed the market for its encryption systems business [1], unlike the rival Swiss firm Crypto AG which, it emerged recently, had effectively been a vehicle for the CIA, initially in conjunction with West German intelligence [10].

ULTIMATE ACHIEVEMENT: SWIFT

The international finance sector was always a large target market, a market that was defined and presided over by a funds transfer banking titan, the Society for Worldwide Interbank Financial Telecommunication (SWIFT). Ran Data put considerable effort into the pursuit of this "jewel in the crown" of banking. Sala and colleagues worked endlessly to demonstrate that the Ran Data encryptors had specifications exceeding those of the competition and were well priced. In 1987 success was finally achieved, with SWIFT placing orders with Ran Data [27]. This, as anticipated, saw other banks follow: SWIFT had 2537 financial customers connected in 1988, many of which became clients of Ran Data.

Sala was made a Member of the Order of Australia (AM) in 1984 in recognition of his pioneering services to the electronics industry [24], [34]. This was most welcome to him, but his real reward was to have taken on the giants of industry in the US and Europe and to have beaten them. The technological excellence that led to Sala's encryption designs and which generated worldwide demand was the recognition that he most cherished. This was the pinnacle of his career. Most of this actually happened after the AM award.

Although Ran Data encryptors in various forms were marketed well into the 1990s [3], Sala decided to leave Ran Data in February 1989, following a restructure of the company's share register and change in the controlling interest brought on by the 1987 stock market crash.

Sala kept busy with a variety other interests until he died on 1 April 2002, publishing with his son a paper on steganography only weeks before his death [26].

MONTE SALA'S LEGACY

Sala is not well-known outside a relatively small local circle. This record of his achievements has been undertaken in an attempt to redress that and to accord him wider recognition. He had always been passionate about and worked and lobbied tirelessly for the establishment of an electronics industry in Western Australia. He was also devoted to fostering links between academia, industry, and government, which he saw worked well in his own case, and he saw it working exceptionally well with NASA. His influence, along with kindred spirits such as the Deputy Premier of Western Australia at the time, contributed to the establishment of the WA Technology Park adjacent to Curtin University in 1985.

Most of this occurred before what are considered to be his most significant and enduring achievements, which were made in the field of encryption.

Sala should be remembered first for the intuitive grasp he possessed of how to solve a wide range of technological, electronic problems, often taking merely days to conceive and design a solution and, in some instances, build a prototype. This was exemplified throughout his electronics career. One of his common expressions was "I can do that!" when presented with a challenge [18]. And this was not false confidence—he always did. This is best illustrated when he was first challenged to develop a general-purpose communications line encryption system—he had a proof of concept designed, built and running on matrix board flawlessly in just 10 days [11].

Second, in the murky world of national and commercial espionage and code cracking, his integrity stood out. He refused to compromise the integrity of his encryption system, albeit some important markets were then closed to him.

Finally, two aspects of his encryption inventions stand out. First, he designed and built a system incorporating genuine random numbers as a basis for lottery draws [30], [33], which was used for years in lottery systems around Australia. Second, the crown in his achievements, the EKES encryption system, deemed to be uncrackable [16], [28], was taken up by many banks but most importantly by SWIFT [27].

Throughout his career, Sala exhibited exceptional intellectual ability and creative power. This record of his achievements has been compiled in the hope that they will be recognized, remembered, and celebrated more widely, as they deserve to be.

ACKNOWLEDGMENTS

The author wishes to acknowledge the significant input that Monte Sala's son, Tony Sala, has provided to the preparation of this article, having drawn extensively on Tony's intimate knowledge of much of the material, and valuing the corrections that Tony has made to the manuscript; the author accepts full responsibility for any remaining errors. The author also wishes to acknowledge the valuable contributions made by Dennis Moore and Terry Woodings in preparing this article.

BIBLIOGRAPHY

[1] The Australian, "Top-secret unit 'told electronics firm to spy,'" Dec. 3, 1988.

[2] D. Burr and J. Ross, "Vision: The world through picket fences," *Curr. Biol.*, vol. 14, no. 10, pp. R381–R382, May 2004, doi: 10.1016/j.cub.2004.05.011.

[3] "Contract PO99340 (30.1.96) for Ran Data securlink encryptor, Department of Foreign Affairs & Trade, contracts arranged," in *Commonwealth of Australia Gazette*, 1996, p. 982 [Issue No. PD6].

[4] "Defence export controls," Australian Department of Defence. Accessed: Jul 20, 2023. [Online]. Available: https://www.defence.gov.au/business-industry/export/controls

[5] P. Dench and A. Gregg, "Carnarvon and Apollo: One giant leap for a small Australian town," Kenthurst, Australia: Rosenberg Publishing, 2010.

[6] J. Ford and T. Griggs, "Ran data's 'electronic envelope' Australian technology: Past, present, future, science and the citizen," *Sci. Amer.*, vol. 251, no. 6, p. A29, Dec. 1984.

[7] B. Hardy, "Back in my day," Information Age, Australian Computer Society, Nov. 17, 2022. Accessed: Jul. 20, 2023. [Online]. Available: https://ia.acs.org.au/content/ia/article/2022/back-in-my-day–brian-hardy.html

[8] J. Jesser, "Ran data's big Europe contract," *Canberra Times*, Computers and Technology, p. 18, May 1985.

[9] McKinsey & Co., "Report on building a successful Ran Data business in the US," prepared for Merrill Lynch Pierce Fenner and Smith, New York, NY, USA, Nov. 1981, [copy held in Tony Sala's private collection].

[10] G. Miller, "The intelligence coup of the century: For decades, the CIA read the encrypted communications of allies and adversaries," Washington Post, Feb. 11, 2020. Accessed: Jul. 20, 2023. [Online]. Available: https://www.washingtonpost.com/graphics/2020/world/national-security/cia-crypto-encryption-machines-espionage/?itid=ap_gregmiller

[11] D. Moore, "Interview with Penny Collings as part of the National Library's 'history of ICT in Australia oral history project," Feb. 7, 2015. Accessed: Jul. 20, 2023. [Online]. Available: http://catalogue.nla.gov.au/Record/6807543?lookfor=ORAL%20TRC%20moore%20%23[format:Audio]&offset=4&max=128; transcript at http://www.alex-reid.com/History/Dennis-Moore-7Feb15-complete-transcript.htm, sections 27 and 28.

[12] NASA Wainwright, Telex to The West Australian newspaper from NASA concerning the telemetry equipment adapted by Monte Sala for Saturn-6 launch, later adopted worldwide by NASA, May 28, 1964.

[13] Ran Data, "Series 183 encryptor," brochure, Ran Data Corp Ltd, Perth, Australia, 1983.

[14] Ran Data, "Series 185 data encryptor: Operation and installation manual," Ran Data Corp Ltd, Perth, Australia, 1987.

[15] T. A. Reid, "Computing," in *Historical Encyclopedia of Western Australia*, J. Gregory and J. Gothard, Eds. Perth, Australia: UWA Press, 2009, p. 224.

[16] G. Riley, "Siromath Pty Ltd: Report on the security of Ran Data's entropic key encrypting system (EKES)," Jul. 4, 1984 [document held in Tony Sala's private collection].

[17] J. Ross, "A new type of display relying on vision's sensitivity to motion," *J. Physiol.*, vol. 271, pp. 2P–3P, 1977.

[18] J. Ross, "Oral history," University of WA Oral Histories Project. Accessed: Jul. 20, 2023. [Online]. Available: http://www.web.uwa.edu.au/uwahs/oral-histories

[19] J. Ross and A. Sala-Spini, "Patent for picture transmission systems, number 3958077, U.S. patent and trademark office," May 18, 1976. Accessed: Jul. 20, 2023. [Online]. Available: https://image-ppubs.uspto.gov/dirsearch-public/print/downloadPdf/3958077

[20] J. Ross and A. Sala-Spini, "Patent for improvements in graphic display systems, number 1976021004, Australian Government Patent Office," Dec. 21, 1976. Accessed: Jul. 20, 2023. [Online]. Available: http://pericles.ipaustralia.gov.au/ols/auspat/applicationDetails.do?applicationNo=1976021004

[21] J. Ross and A. Sala-Spini, "Patent for the moving graphic display system, number 1985046526, Australian government patent office," Aug. 20, 1985. Accessed: Jul. 20, 2023. [Online]. Available: http://pericles.ipaustralia.gov.au/ols/auspat/applicationDetails.do?applicationNo=1985046526

[22] A. Sala, "A Sala—Sales achievements," Internal Deltec Pty Ltd memo detailing sales to 16 organisations across Australia in 1976 totalling over $5 million in revenue [document held in Tony Sala's private collection].

[23] A. Sala, "Letter as a director of Deltec to Professor R F Whelan, Vice-Chancellor," University of Western Australia, Perth, Australia, May 24, 1976.

[24] A. Sala, "A biographical account between 1950 and 1983," Citation for being made a Member of the Order of Australia, Jun. 1984.

[25] A. Sala, "The moon had to wait: The Odyssey from the sewers to the stars of an Istro-Dalmatian migrated to Australia in 1950," Milan, Spring 1989; J S Battye Library of West Australian History, MN 1780, Papers of Amedeo Sala, ACC 5350A, 1998.

[26] A. Sala and T. U. Sala, "La codifica delle foto (image coding), Bollettino della Comunità Scientifica in Australasia," in *Ufficio Dell'addetto Scientifico*, Deakin, Australia: Italian Embassy, 2002, pp. 97–100.

[27] T. Sala, "Electronic enigmas defend data," *SIBOS* (SWIFT International Banking Operations Seminar), pp. 78–79, 1988. Accessed: Jul. 20, 2023. [Online]. Available: https://www.sibos.com/ [but does not have historical content]. [copy of this paper held in Tony Sala's private collection].

[28] A. Shamir, "Report to Ran Data on the efficacy of the EKES encryption system," Jun. 01, 1984, [document held in Tony Sala's private collection].

[29] H. Shephard, "Report for Ran Data Pty ltd on the picture scrambling system," 1981, [document held in Tony Sala's private collection].

[30] "Casket draw in 3 minutes," *Telegraph*, p. 3, Jun. 12, 1978.

[31] Newsheet of the University of Western Australia, "The Code Breaker," *UniNews*, vol. 2, no. 11, p. 1, Jul. 18, 1983.

[32] Circulated within the University of Western Australia, "The inventors and the Betagraph," *Univ. News*, vol. 7, no. 1, p. 1, Mar. 1976.

[33] University of Western Australia, "Improved random noise generator," Australian Patent Application, Mar. 19, 1979.

[34] Wikipedia, "1984 queen's birthday honours (Australia)," Accessed: Jul. 20, 2023. [Online]. Available: https://en.wikipedia.org/wiki/1984_Queen%27s_Birthday_Honours_(Australia)

[35] World Fair for Technology Exchange, Chicago, February, 1976. Australian Trade Commission (1975-10-02), "Trade promotion diary, show your product to overseas buyers," *Overseas Trading*, vol. 27, no. 19, p. 463, Oct. 1975.

T. ALEX REID is an honorary professorial fellow at the University of Western Australia, Crawley, WA 6009, Australia. Contact him at alex.reid@uwa.edu.au.

EDITOR: Colette Perold, colette.perold@colorado.edu

DEPARTMENT: THINK PIECE

This article originally appeared in IEEE Annals of the History of Computing vol. 45, no. 4, 2023

Dissecting Data: History of Data as History of the Body

Andrew S. Lea, *Brigham and Women's Hospital, Harvard Medical School, Boston, MA, 02115, USA*

I met my cadaver before I had met most of my classmates. Like many medical students, I was glad to have anatomy as the first block of medical school. But my reasons were different: Always a bit squeamish, I figured it was better to discover if I was not cut out for medicine early on—before I was in too deep. Over seven weeks, we systematically dissected our assigned cadaver, dutifully following each blade stroke as spelled out in *Grant's Dissector*. We started with the heart. The face, covered by a damp cloth during the earlier stages of dissection, was saved for the last day. The ostensible reason for this concealment was utilitarian: To prevent the face from desiccating. But I suspect its more significant function, if not aim, was that of depersonalization.

At the end of the course, we learned more about the person behind the cadaver: How they died and how they ended up on the dissecting table. While some had earmarked their bodies for donation to the anatomy lab in advance of dying, many cadavers today are "donors by circumstance"—unclaimed bodies from people, often impoverished, unhoused, and dispossessed, who were not claimed by any living relatives or associates. Ours was among the unclaimed. In Maryland, bodies fall under control of the State Anatomy Board and can become "donors by circumstance" if they remain unclaimed 72 h following a "reasonable search."[1]

The ethics of this situation has been thoroughly analyzed and rightfully critiqued.[2] Discomfort with the use of unclaimed bodies in anatomy laboratories has prompted many institutions to turn to new digital tools for teaching anatomy. A growing number of online applications and programs permit learning anatomy through engagement with "digital bodies" on smart phones and tablets. Many feel reassured using these tools. (Surely these digital bodies circumvent the problems of respect for persons and consent?) But in fact these digital bodies came from people: Real human bodies furnished the data on which digital anatomical tools rely. At the Francis A. Countway Library of Medicine in Boston, Harvard medical students can learn anatomy using the Anatomage Table, an interactive digital atlas based on cross-sectional anatomical, CT, and MRI images in the public domain. The male body displayed by the Anatomage Table is based on data from Joseph Paul Jernigan, a 39-year-old Texas man who was executed by lethal injection in 1993.[3] Although Jernigan nominally gave consent for the use of his data, one wonders whether consent can truly be given under conditions of capital punishment.[4]

Cadaveric dissection offers an especially clear example of the entanglements of data with bodies, but the relationship can be further explored. In this essay, I sample works at the intersection of the history of data and the history of the body and lay out an agenda for bringing the fields closer together. I outline two ways of thinking about the history of data through a history of the body lens: Data having bodies, and data coming from bodies. Putting the history of data into more direct conversation with the rich scholarship on the history of the body not only yields important insights into the history and nature of data but also places the ethical stakes around the present day uses of data into sharper focus.

DATA AS BODIES

The challenge of learning anatomy is principally one of volume: The sheer number of facts to memorize.

[1]MD Health-Gen Code, 2013.
[2]D.G. Jones and M.I. Whitaker, 2012.

[3]K. Keet and B. Kramer, 2022.
[4]S. Hildebrandt, 2008.

1058-6180 © 2023 IEEE
Digital Object Identifier 10.1109/MAHC.2023.3326052
Date of current version 23 November 2023.

In between sessions at the dissection bay, I did my best to cram my brain with thousands of facts about human bodies. Most medical students had developed some material system of managing this data deluge. Many turned to Anki, a then-new spaced-repetition application, whose algorithm would present flashcards to learners at varying intervals depending on their familiarity with the cards on prior iterations. My own system was manual. I created hundreds of paper flashcards that I sorted into different boxes depending on how well I knew the card. By the end of the seven weeks my deck of cards had seen better days. The corners were bent, the edges torn. On many cards, the ink had smudged to the point of illegibility.

Recent years have seen growing attention directed to the materiality of data. Many STS scholars and historians have shown how data exist in the material world; they are incarnate in stuff, whether notebooks, spreadsheets, punched cards, or magnetic tape. As part of this material turn, many have looked to the physical sciences for rich metaphors to better understand the nature and function of data. In his groundbreaking study on climate modeling, Paul Edwards invoked the notion of "data friction" to understand the materiality of data and their sociotechnical kinetics. "Data are *things*," Edwards writes. "They are not just numbers but also numerals, with dimensionality, weight, and texture. "Data friction" refers to the costs in time, energy, and attention required simply to collect, check, store, move, receive, and access data."[5] The invocation of data having friction implies that it also has mass. (Mass is, after all, one of the variables that goes into calculating friction.) This insight has spurred other fruitful Newtonian metaphors, such as framing the epistemologies and biases baked into a dataset as a kind of "data inertia."[6]

What, then, is gained by a shift in metaphor from physics to physiology? Seeing the corporality of data creates opportunities to draw on analytic tools from the history of the body. First, the data-as-bodies metaphor foregrounds the materiality of data. Dealing with the body can be a messy affair; bodies leak, swell, crack, desiccate, and smell. Data can be just as messy. Second, like bodies, data depend on maintenance and care: data must be "cleaned," code must be "debugged," and software must be updated. Third, bodies are complex systems, irreducible to the function of their individual components. Such a perspective invites exploration of the emergent and entangled properties of datasets—how they depend on and operate within a complex system of infrastructures, regulations, human actors, social mores, and much more. Finally, the history of the body also offers productive conceptual resources for analyzing and critiquing anthropomorphic discourses around artificial "intelligence."

MANY STS SCHOLARS AND HISTORIANS HAVE SHOWN HOW DATA EXIST IN THE MATERIAL WORLD; THEY ARE INCARNATE IN STUFF, WHETHER NOTEBOOKS, SPREADSHEETS, PUNCHED CARDS, OR MAGNETIC TAPE.

DATA FROM BODIES

The data on my anatomy flashcards also came *from* bodies. Some of this information came directly from engagements with my assigned cadaver. But other flashcards contained systematized knowledge generated through anatomists' interactions with past bodies. The process by which bodies are made into data has often been extractive, playing out in the contexts of capitalism and colonialism.[7] But the process can also be empowering, such as when disenfranchised groups create databases to ensure that their bodies and lives are represented in the datasets that drive medical knowledge and practice.

Many historians have drawn attention to the human origins of data. Iris Clever, for example, has studied the mountains of data amassed by physical anthropologists as they measured human bodies, bones, and skulls.[8] Rebecca Lemov has chronicled the making of "personal" data through the history of Don Talayesva, a Hopi Indian who gave his personal materials to scientists for the creation of a "vast data set" in the mid-twentieth century.[9] The bodies behind datasets are sometimes harder to make out. The historian Joanna Radin, for example, has traced the career of the Pima Indian Diabetes Dataset (PIDD), whose origins from American Indian bodies were sometimes forgotten (or ignored) as the

[5]P. Edwards, 2010, 84.
[6]A. Lea, 2019.
[7]T. Kukutai and J. Taylor, 2016.
[8]I. Clever, 2023.
[9]R. Lemov, 2017.

dataset was harnessed for all kinds of medical and machine learning research.[10]

The work of *making* data is also an embodied process. Despite the word's etymology, "data" are not given or bestowed; they are made. As Geoffrey Bowker has noted, there is no such thing as "raw" data; data are, in one way or another, always already "cooked"—collected, cleaned, categorized, cared for.[11] This process of creating data is laborious. In the mid-twentieth century, physicians working to create databases to help with diagnosis described the work as "a slow, tiresome, and troubling job."[12] A history-of-the-body framing underscores the embodied labor of collecting, making, cleaning, processing, and sorting data. These processes demand work in the most corporeal sense: they break the back, strain the eyes, corrode the joints. Data, then, come from bodies in more than one way. They represent information about human bodies. But they also exist thanks to the frequently unrecognized embodied labor of data workers.

CONCLUSION: THE ANATOMY OF DATA

The conceptual synergy between data and body opens up many avenues for study beyond those outlined in this essay. The association, for instance, invites further exploration of the connections between the history of data and other disciplines where the body has long been a central and productive focus: From disability studies to queer and gender studies to carceral studies. The recent volume *Abstractions and Embodiments*, edited by Janet Abbate and Stephanie Dick, fleshes out many of these interdisciplinary connections by "foregrounding bodies as the site where the power and pain of computer technology play out."[13] Above all, seeing data as and from bodies—that is, understanding "the anatomy of data"—allows for a more considered, ethical engagement with the data we use and had a hand in making.

BIBLIOGRAPHY

[1] J. Abbate and S. Dick, "Introduction: Thinking with computers," in *Abstractions and Embodiments: New Histories of Computing and Society*, J. Abbate and S. Dick, Eds., Baltimore, MD, USA: The Johns Hopkins Univ. Press, 2022, Art. no. 11.

[2] B. Bjerregaard et al., "Computer-aided diagnosis of the acute abdomen: A system from Leeds used on Copenhagen patients," in *Decision Making and Medical Care*, F. de Dombal and F. Gremny, Eds. Amsterdam, The Netherlands: North Holland, 1976.

[3] G. Bowker, *Memory Practices in the Sciences*. Cambridge, MA, USA: MIT Press, 2005.

[4] I. Clever, "Biometry against fascism: Geoffrey Morant, race, and anti-racism in twentieth-century physical anthropology," *Isis*, vol. 114, no. 1, pp. 25–49, 2023.

[5] P. Edwards, *A Vast Machine: Computer Models, Climate Data, and the Politics of Global Warming*. Cambridge, MA, USA: MIT Press, 2010, Art. no. 84.

[6] S. Hildebrandt, "Capital punishment and anatomy: History and ethics of an ongoing association," *Clin. Anatomy*, vol. 21, no. 1, pp. 5–14, 2008.

[7] D. G. Jones and M. I. Whitaker, "Anatomy's use of unclaimed bodies: Reasons against continued dependence on an ethically dubious practice," *Clin. Anatomy*, vol. 25, no. 2, pp. 246–254, Mar. 2012.

[8] K. Keet and B. Kramer, "Advances in digital technology in teaching human anatomy: Ethical predicaments," in *Biomedical Visualisation*, L. Shapiro and P. M. Rea, Eds. Berlin, Germany: Springer, 2022.

[9] T. Kukutai and J. Taylor, *Indigenous Data Sovereignty: Toward an Agenda*. Canberra, Australian: ANU Press, 2016.

[10] A. Lea, "Computerizing diagnosis: Keeve Brodman and the medical data screen," *ISIS*, vol. 110, no. 2, pp. 228–249, 2019.

[11] R. Lemov, "Anthropology's most documented man, ca. 1947: A prefiguration of Big Data from the big social science era," *Osiris*, vol. 32, no. 1, pp. 21–42, 2017.

[12] MD Health-Gen Code § 5-406, 2013.

[13] J. Radin, "Digital natives: How medical and indigenous histories matter for Big Data," *Osiris*, vol. 32, no. 1, pp. 43–64, 2017.

ANDREW S. LEA is a resident physician in internal medicine at Brigham and Women's Hospital and a clinical fellow at Harvard Medical School, Boston, MA, 02115, USA. His first book, *Digitizing Diagnosis: Medicine, Minds, and Machines in Twentieth-Century America*, was recently published by Johns Hopkins University Press. Contact him at alea@bwh.harvard.edu.

[10] J. Radin, 2017.
[11] G. Bowker, 2005.
[12] B. Bjerregaard et al., 1976.
[13] J. Abbate and S. Dick, 2022, 11.

IEEE COMPUTER SOCIETY D&I FUND

Drive Diversity & Inclusion in Computing

Supporting projects and programs that positively impact diversity, equity, and inclusion throughout the computing community.

DONATE TODAY!

IEEE COMPUTER SOCIETY

IEEE Foundation

CALL FOR SPECIAL ISSUE PROPOSALS

Computer solicits special issue proposals from leaders and experts within a broad range of computing communities. Proposed themes/issues should address important and timely topics that will be of broad interest to *Computer*'s readership. Special issues are an essential feature of *Computer*, as they deliver compelling research insights and perspectives on new and established technologies and computing strategies.

Please send us your high-quality proposals for the 2025–2026 editorial calendar. Of particular interest are proposals centered on:

- 3D printing
- Robotics
- LLMs
- AI safety
- Dis/Misinformation
- Legacy software
- Microelectronics

Proposal guidelines are available at:

www.computer.org/csdl/magazine/co/write-for-us/15911

IEEE COMPUTER SOCIETY

IEEE

stay connected.

Join our online community! Follow us to stay connected wherever you are:

- @ComputerSociety
- facebook.com/IEEEComputerSociety
- IEEE Computer Society
- youtube.com/IEEEComputerSociety
- instagram.com/ieee_computer_society

IEEE COMPUTER SOCIETY

IEEE

IEEE

SECURITY & PRIVACY

IEEE Security & Privacy **is a bimonthly magazine communicating advances in security, privacy, and dependability** in a way that is useful to a broad section of the professional community.

The magazine provides articles with both a practical and research bent by the top thinkers in the field of security and privacy, along with case studies, surveys, tutorials, columns, and in-depth interviews. Topics include:

- Internet, software, hardware, and systems security
- Legal and ethical issues and privacy concerns
- Privacy-enhancing technologies
- Data analytics for security and privacy
- Usable security
- Integrated security design methods
- Security of critical infrastructures
- Pedagogical and curricular issues in security education
- Security issues in wireless and mobile networks
- Real-world cryptography
- Emerging technologies, operational resilience, and edge computing
- Cybercrime and forensics, and much more

www.computer.org/security

Join the IEEE Computer Society for subscription discounts today!

www.computer.org/product/magazines/security-and-privacy

IEEE COMPUTER SOCIETY

IEEE

IEEE TRANSACTIONS ON

COMPUTERS

Call for Papers

Publish your work in the IEEE Computer Society's flagship journal, *IEEE Transactions on Computers* (*TC*). *TC* is a monthly publication with a wide distribution to researchers, industry professionals, and educators in the computing field.

TC seeks original research contributions on areas of current computing interest, including the following topics:

- Computer architecture
- Software systems
- Mobile and embedded systems
- Security and reliability
- Machine learning
- Quantum computing

All accepted manuscripts are automatically considered for the monthly featured paper and annual Best Paper Award.

Learn about calls for papers and submission details at
www.computer.org/tc

IEEE COMPUTER SOCIETY

IEEE

Get Published in the *IEEE Open Journal of the Computer Society*

Get more citations by publishing with the *IEEE Open Journal of the Computer Society*

Your research on computing and informational technology will benefit from 5 million unique monthly users of the *IEEE Xplore*® Digital Library. Plus, this journal is fully open and compliant with funder mandates, including Plan S.

Submit your paper today!
Visit www.computer.org/oj to learn more.

IEEE COMPUTER SOCIETY | IEEE

IEEE TRANSACTIONS ON

BIG DATA

***IEEE Transactions on Big Data* is a quarterly journal that publishes peer-reviewed articles with big data as the main focus.**

The articles provide cross-disciplinary, innovative research ideas and applications results for big data including novel theory, algorithms, and applications. Research areas include:

- Big data
  - Analytics
  - Curation and management
  - Infrastructure
  - Performance analyses
  - Semantics
  - Standards
  - Visualization
- Intelligence and scientific discovery from big data
- Security, privacy, and legal issues specific to big data

Applications of big data in the fields of endeavor where massive data is generated are of particular interest.

www.computer.org/tbd

Join the IEEE Computer Society for subscription discounts today!

www.computer.org/product/journals/tbd

IEEE COMPUTER SOCIETY

IEEE

Conference Calendar

IEEE Computer Society conferences are valuable forums for learning on broad and dynamically shifting topics from within the computing profession. With over 200 conferences featuring leading experts and thought leaders, we have an event that is right for you. Questions? Contact conferences@computer.org.

MAY

4 May

- ARITH (IEEE Symposium on Computer Arithmetic), El Paso, USA
- FCCM (IEEE Annual Int'l Symposium on Field-Programmable Custom Computing Machines), Fayetteville, USA
- MOST (IEEE Int'l Conf. on Mobility, Operations, Services and Technologies), Newark, USA

5 May

- CAI (IEEE Conf. on Artificial Intelligence), Santa Clara, USA
- HOST (IEEE Int'l Symposium on Hardware Oriented Security and Trust), San Jose, USA

6 May

- RTAS (IEEE Real-Time and Embedded Technology and Applications Symposium), Irvine, USA

11 May

- ASYNC (IEEE Int'l Symposium on Asynchronous Circuits and Systems), Portland, USA
- ISPASS (IEEE Int'l Symposium on Performance Analysis of Systems and Software), Ghent, Belgium

12 May

- SP (IEEE Symposium on Security and Privacy), San Francisco, USA

19 May

- CCGrid (IEEE Int'l Symposium on Cluster, Cloud and Internet Computing), Tromsø, Norway
- ICDE (IEEE Int'l Conf. on Data Eng.), Hong Kong
- ICFEC (IEEE Int'l Conf. on Fog and Edge Computing) Tromsø, Norway

26 May

- FG (IEEE Int'l Conf. on Automatic Face and Gesture Recognition), Tampa/Clearwater, USA

27 May

- WoWMoM (IEEE Int'l Symposium on a World of Wireless, Mobile and Multimedia Networks), Fort Worth, USA

JUNE

2 June

- MDM (IEEE Int'l Conf. on Mobile Data Management), Irvine, USA

3 June

- IPDPS (IEEE Int'l Parallel and Distributed Processing Symposium), Milano, Italy

5 June

- ISMVL (IEEE Int'l Symposium on Multiple-Valued Logic), Montreal, Canada

9 June

- DCOSS-IoT (Int'l Conf. on Distributed Computing in Smart Systems and the Internet of Things), Lucca, Italy
- SKIMA (Int'l Conf. on Software, Knowledge, Information Management & Applications), Paisley, United Kingdom

11 June

- CVPR (IEEE/CVF Conf. on Computer Vision and Pattern Recognition), Nashville, USA

16 June

- CSF (IEEE Computer Security Foundations Symposium), Santa Cruz, USA

18 June

- CBMS (IEEE Int'l Symposium on Computer-Based Medical Systems), Madrid, Spain
- ICHI (IEEE Int'l Conf. on Healthcare Informatics), Rende, Italy

21 June

- ISCA (ACM/IEEE Annual Int'l Symposium on Computer Architecture), Tokyo, Japan

23 June

- DSN (Annual IEEE/IFIP Int'l Conf. on Dependable Systems and Networks), Naples, Italy
- SVCC (Silicon Valley Cybersecurity Conf.), San Francisco, USA

26 June

- IEEE Cloud Summit, Washington, DC, USA

 2469-7087/25 © 2025 IEEE

- MARIS (Symposium on Maritime Informatics and Robotics), Syros, Greece

30 June

- EuroS&P (IEEE European Symposium on Security and Privacy), Venice, Italy
- ICME (IEEE Int'l Conf. on Multimedia and Expo), Nantes, France

JULY

2 July

- ISCC (IEEE Symposium on Computers and Communications), Bologna, Italy

6 July

- ISVLSI (IEEE Computer Society Annual Symposium on VLSI), Kalamata, Greece

7 July

- IOLTS (IEEE Int'l Symposium on On-Line Testing and Robust System Design), Ischia, Italy
- SERVICES (IEEE World Congress on Services), Helsinki, Finland

8 July

- COMPSAC (IEEE Annual Computers, Software, and Applications Conf.), Toronto, Canada

14 July

- ICALT (IEEE Int'l Conf. on Advanced Learning Technologies), Changhua, Taiwan

20 July

- ICDCS (IEEE Int'l Conf. on Distributed Computing Systems), Glasgow, United Kingdom

21 July

- CISOSE (IEEE Int'l Congress on Intelligent and Service-Oriented Systems Eng.), Tucson, USA
- ICCP (IEEE Int'l Conf. on Computational Photography), Toronto, Canada

28 July

- ASAP (IEEE Int'l Conf. on Application-specific Systems, Architectures and Processors), Vancouver, Canada
- SCC (IEEE Space Computing Conf.), Los Angeles, USA
- SMC-IT (IEEE Int'l Conf. on Space Mission Challenges for Information Technology), Pasadena, USA

AUGUST

1 August

- PCDS (IEEE Int'l Conf. on Privacy Computing and Data Security), Hakodate, Japan

4 August

- COINS (IEEE Int'l Conf. on Omni-layer Intelligent Systems), Madison, USA

6 August

- IRI (IEEE Int'l Conf. on Information Reuse and Integration and Data Science), San Jose, USA

11 August

- FiCloud (Int'l Conf. on Future Internet of Things and Cloud), Istanbul, Turkiye

20 August

- HOTI (IEEE Symposium on High-Performance Interconnects), virtual
- RTCSA (IEEE Int'l Conf. on Embedded and Real-Time Computing Systems and Applications), Singapore

24 August

- HCS (IEEE Hot Chips 37 Symposium), Stanford, USA,

26 August

- PST (Annual Int'l Conf. on Privacy, Security, and Trust), Fredericton, Canada

SEPTEMBER

1 September

- RE (IEEE Int'l Requirements Eng. Conf.), Valencia, Spain

2 September

- CLUSTER (IEEE Int'l Conf. on Cluster Computing), Edinburgh, United Kingdom

IEEE COMPUTER SOCIETY CAREER CENTER

Career Accelerating Opportunities

Explore new options—upload your resume today

careers.computer.org

Changes in the marketplace shift demands for vital skills and talent. The **IEEE Computer Society Career Center** is a valuable resource tool to keep job seekers up to date on the dynamic career opportunities offered by employers.

Take advantage of these special resources for job seekers:

- JOB ALERTS
- TEMPLATES
- WEBINARS
- CAREER ADVICE
- RESUMES VIEWED BY TOP EMPLOYERS

No matter what your career level, the IEEE Computer Society Career Center keeps you connected to workplace trends and exciting career prospects.

IEEE COMPUTER SOCIETY

IEEE