COMPUTING edge

- Security and Privacy
- Software Engineering
- Smart Homes
- Ethics

APRIL 2023

www.computer.org

IEEE COMPUTER SOCIETY

IEEE

IEEE CAI

2023 IEEE Conference on Artificial Intelligence

NEW EVENT

5–6 June 2023

Hyatt Regency, Santa Clara, California, USA

Exclusively In-Person

REGISTER TODAY

Be a part IEEE CAI—an exciting two-day event covering the applications of AI and key AI verticals that impact industrial technology and innovations.

At IEEE CAI you'll gain insights from over 30 leading artificial intelligence innovators on the following topics:

- AI in Healthcare/ Life Sciences
- AI in Transportation/ Aerospace
- AI in Energy
- Industrial AI
- AI for Earth Systems Decision Support
- Ethical & Societal Implications of AI

https://cai.ieee.org/2023

PLATINUM SUPPORTER

QUANTINUUM

SILVER SUPPORTER

IEEE Computational Intelligence Society

IEEE COMPUTER SOCIETY

IEEE SMC Systems, Man, and Cybernetics Society

IEEE Signal Processing Society

IEEE

COMPUTING edge

IEEE COMPUTER SOCIETY computer.org

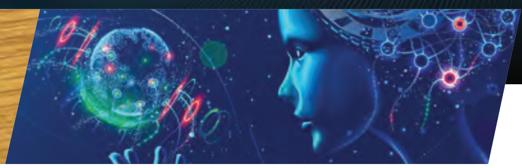
IEEE COMPUTER SOCIETY   IEEE   Printed with inks containing soy and/or vegetable oils

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.

APRIL 2023 · VOLUME 9 · NUMBER 4

COMPUTING edge



Security and Privacy

Software Engineering

Smart Homes

Ethics

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

Changing Landscape of Technical Education Pedagogy From Traditional to Practical e-Learning

With the COVID-19 pandemic, online university education assumed greater importance. The authors of this article from the November 2022 issue of *Computer* propose a practical e-learning model with suggestions to augment online education of all forms to support effective technical education.

Computing in Science & Engineering

MolSSI Education: Empowering the Next Generation of Computational Molecular Scientists

The Molecular Sciences Software Institute (MolSSI) is a research and education center that supports software development in the computational molecular sciences (CMS). One of MolSSI's core objectives is to provide education and training for the next generation of computational researchers. MolSSI Education targets various career stages and skill levels through its live workshops, online resources, and software fellowship program. This article from the May/June 2022 issue of *Computing in Science & Engineering* delineates educational efforts at the MolSSI, overall goals, and resources that can be useful to researchers in CMS.

IEEE Annals of the History of Computing

The Present of the Past: A Sociotechnological Framework for Understanding the Availability of Research Materials

Research relies on the survival of materials about activities of everyday life. This article outlines a sociotechnological framework for the social life course of research materials, through capturing, saving, and retrieving, and then applies this approach to understand the survival rates of historical demographic materials, censuses, and census-like surveys. This approach emphasizes that research materials survive because of (or despite) not only technical elements, but also interconnected sociotechnological processes. These processes are shaped by the power and interests of social and state actors and institutions. Read more in this article from the October–December 2022 issue of *IEEE Annals of the History of Computing*.

IEEE Computer Graphics and Applications

Which Biases and Reasoning Pitfalls Do Explanations Trigger? Decomposing Communication Processes in Human–AI Interaction

Collaborative human–AI problem-solving and decision-making rely on effective communications between both agents. Such communication processes comprise explanations and interactions between a sender and a receiver. Investigating these dynamics is crucial to avoid miscommunication problems. In this article from the November/December 2022 issue of *IEEE Computer Graphics and Applications*, the authors propose a communication dynamics model, examining the impact of the sender's explanation intention and strategy on the receiver's perception of explanation effects. They present

2469-7087/23 © 2023 IEEE

potential biases and reasoning pitfalls with the aim of contributing to the design of hybrid intelligence systems. Finally, they propose six desiderata for human-centered explainable AI and discuss future research opportunities.

IEEE Intelligent Systems

CACLA-Based Local Path Planner for Drones Navigating Unknown Indoor Corridors

This *IEEE Intelligent Systems* September/October 2022 article presents an online local path planning approach for autonomous drones navigating a 2D plane in an unknown, indoor corridor-like environment. The proposed method utilizes a reinforcement learning approach for training a local path planner for navigation in said environment. With a continuous actor-critic learning automaton (CACLA) applied for continuous action spaces, the proposed algorithm uses a reward structure that formulates a balancing function that gives reward based on balancing the vehicle between artificial potential hills. The drone thereby learns steering control and obstacle avoidance while maintaining a central aligned position with respect to the unknown hallways or corridors.

IEEE Internet Computing

Empowering Citizens With Digital Twins: A Blueprint

The exponentially growing amount of digital information and data analysis increase the ability to perceive the holistic situation of people. This article from *IEEE Internet Computing*'s September/October 2022 issue applies the digital twin paradigm to strengthen a person's ability to utilize information about themselves by creating a digital representation of their situation to support their well-being. The authors propose a blueprint to empower individuals by improving their self-determination regarding their personal data. The blueprint will help service and data providers, both public and private, to develop a common understanding of the role and possibilities of a citizen's controlled personal digital twin of themselves for creating people-centric solutions.

IEEE micro

Increasing Throughput of In-Memory DNN Accelerators by Flexible Layerwise DNN Approximation

Approximate computing and mixed-signal in-memory accelerators are promising paradigms to significantly reduce computational requirements of deep neural network (DNN) inference without accuracy loss. In this November/December 2022 *IEEE Micro* article, the authors present a novel in-memory design for layer-wise approximate computation at different approximation levels. A sensitivity-based high-dimensional search is performed to explore the optimal approximation level for each DNN layer. The new methodology offers high flexibility and optimal tradeoff between accuracy and throughput, which is demonstrated by an extensive evaluation on various DNN benchmarks for medium- and large-scale image classification with CIFAR10, CIFAR100, and ImageNet.

IEEE MultiMedia

Adversarial Adaptive Interpolation in Autoencoders for Dually Regularizing Representation Learning

Linear interpolation in the latent space may induce mismatch between the constructed data and the distribution on which a model was trained. In this article from *IEEE MultiMedia*'s July–September 2022 issue, the authors propose an Adversarial Adaptive Interpolation-based AutoEncoder

(AdvAI-AE). To constrain the interpolation path on the underlying manifold, an additional interpolation correction module is trained to offset the deviation between the linearly interpolated data points and the statistics of real ones in latent space. The authors apply prior matching to control the characteristics of the representation. Toward this end, the maximum mean discrepancy-based and adversarial regularizers are incorporated into the model. The synthesized data from random variables are in turn leveraged to regularize the interpolation process.

IEEE Pervasive Computing

Citizen Manufacturing: Unlocking a New Era of Digital Innovation

We have all come to expect—if not depend upon—the steady march of technology. All manner of pervasive computing devices, applications, and services increasingly support us at home and work. Thankfully, for those tasked with creating future generations of innovative digital technologies, the design and prototyping process continues to get easier. But for hardware, the transition from device prototype to production presents a bottleneck that is restricting the rate and nature of innovation. Imagine instead a world of citizen manufacturing, where individuals are empowered to not only turn their ideas into working prototypes, but also evolve them organically and seamlessly into viable products. Such an approach could increase consumer choice, grow local and national economies, and facilitate more socially conscious production. Read more in this article from *IEEE Pervasive Computing*'s July–September 2022 issue.

IEEE Security & Privacy

Dirty Metadata: Understanding A Threat to Online Privacy

People have a certain expectation of privacy when using widespread and trusted online services. This *IEEE Security & Privacy* article from the November/December 2022 issue surveys several popular web applications to understand if uploaded user images are handled securely using open-source digital forensics tools and a custom software framework.

IEEE Software

Decentralizing Infrastructure as Code

Infrastructure as code (IaC) automates deployments for single teams, falling short of decentralized deployments across groups. We need mature IaC solutions that embrace and consolidate software engineering principles to enable testing and automation advances for decentralized organizations. Read more in this article from the January/February 2023 issue of *IEEE Software*.

IT Professional

Explainable Artificial Intelligence for Smart Grid Intrusion Detection Systems

A popular approach to overcome the complexity of cybersecurity and sophistication of cyberattacks is implementing AI-based security controls that integrate machine learning (ML) algorithms into security controls, such as intrusion and malware detection. These AI-based security controls are considered more effective than traditional signature-based and heuristics-based controls. However, the growing adoption of advanced ML algorithms is turning AI-based security controls into black-box systems. The authors of this September/October 2022 *IT Professional* article postulate that these black-box AI methods would make risk management and informed decision-making challenging.

Ensuring Secure Software

Everyone agrees on the importance of secure software, and yet integrating security into the development of software systems can be challenging. Software engineers need tools, strategies, and standards that prioritize security. This *ComputingEdge* issue highlights innovative work being done to promote a future of secure software.

The authors of "A Research Road Map for Building Secure and Resilient Software-Intensive Systems," from *IEEE Security & Privacy*, contend that the security and software communities must work together to establish development and architectural paradigms for better software engineering processes. *IT Professional*'s "Trust Considerations in Open Banking" describes financial SaaS ecosystems developed under robust security guidelines and regulations.

Security isn't the only concern when striving to develop high-quality software. *IEEE Software*'s "James Smith on Software Bugs and Quality" presents an interview with a bug-catching expert and discusses measuring, benchmarking, and fixing design flaws based on data. The author of "Inclusivity Bugs and the Language We Use," from *Computing in Science & Engineering*, posits that certain terms and phrases that are common in software engineering contexts can be unnecessarily exclusive.

Next, two articles discuss new ideas in smart homes. "Smart Homes or Real Homes: Building a Smarter Grid With 'Dumb' Houses," from *IEEE Pervasive Computing*, proposes a prototype system for monitoring device energy use in houses with modest and non-standardized smart home capabilities. *Computer*'s "Alexa, M.D." demonstrates how intelligent personal assistants, which are often found in smart homes, can also be used to train healthcare professionals.

This *ComputingEdge* issue concludes with two articles on artificial intelligence (AI) ethics. *IEEE Internet Computing*'s "Ethical Online AI Systems Through Conscientious Design" introduces an approach to governance of online hybrid communities of humans and artificial entities. *Computer*'s "Assessing AI Fairness in Finance" provides pragmatic guidance to help banks implement AI systems ethically.

This article originally appeared in IEEE SECURITY & PRIVACY vol. 19, no. 6, 2021

A Research Road Map for Building Secure and Resilient Software-Intensive Systems

Robert Cunningham, *Carnegie Mellon University Software Engineering Institute and University of Pittsburgh*

Anita D. Carleton and Tom Longstaff, *Carnegie Mellon University Software Engineering Institute*

Forrest J. Shull, *Carnegie Mellon University Software Engineering Institute*

Poor software engineering processes can result in insecure and brittle software-intensive systems. A new U.S. agenda addresses this by advancing development and architectural paradigms, and by providing concrete research and development recommendations. We propose that the security community works closely with the software engineering community to realize secure, resilient software-intensive systems.

Many software vulnerabilities arise from poor software engineering processes. Sometimes, they arise during the development process itself, as is the case with the recent SolarWinds attack that provided threat actors with access to thousands of companies and government agencies. To accomplish this supply chain attack, the SolarWinds development environment was breached, and malicious software was inserted into widely deployed network-and infrastructure-monitoring software.[1,2]

Occasionally, the vulnerabilities arise in the software architectural paradigm, especially for extremely large-scale and rapidly changing systems and those that interact with the physical world. Architectural vulnerabilities are sometimes external to the system itself—the vulnerability can be in society and exploited using functionality or services provided by the software architecture.

One such vulnerability is the believability of "fake news" when it favors a candidate that a democratic citizen supports. Large-scale social networks can amplify false information and have done so: in the United States in 2016, the most popular fake news stories were more widely shared on Facebook than the most popular mainstream news stories.[3]

To address this and broader problems, desiderata of human-focused, societal-scale software systems, such as freedom from secret agendas, freedom to access data for good, freedom from censorship, freedom to understand the logic of the machine, and freedom to stay human, have recently been proposed.[4] Broad agreement on principles has not yet occurred, nor have software engineering techniques been developed to achieve and guarantee those or similar freedoms. Research is needed to establish those principles. To address these challenges and more, a new U.S. National Agenda charts a course toward better software engineering by advancing development and architectural paradigms.[5]

BACKGROUND

At the start of every decade this century, the software engineering community has come together to establish a common understanding of earlier results[6] and identify the most important research directions, each time holding a workshop and writing a report. In 2001, the U.S. Interagency Working Group's Software Design and Productivity Coordinating Group

Digital Object Identifier 10.1109/MSEC.2021.3105876
Date of current version: 28 October 2021

INTRODUCING *IEEE SECURITY & PRIVACY'S* "GLOBALLY SPEAKING" COLUMN

Welcome to the first edition of "Globally Speaking!" The idea for this column came from a conversation with our editor in chief. Over the course of a long career, I have been—in turn—fascinated, startled, delighted, and occasionally appalled by the state of security and privacy around the globe. While there is a surprising amount of collaboration, there are still far too many stovepipes. While a few stovepipes are necessary, allowing nations and organizations to protect populations, assets, and values, many others are counterproductive. There is a need to step back and consider where there are global problems that need to be better understood and worked on by all. Therefore, we brainstormed "Globally Speaking" to provide a forum to consider security and privacy from a global perspective. Appropriate columns will include those that point out barriers that constrain our thinking, elevate policies and practices that either help or hurt global collaboration, dive deeply into critical technologies, or otherwise elevate issues with global impact. The columns will be both by invitation and by submission. We will welcome practitioners, policy makers, researchers, and those affected by global issues. This inaugural column is an invited discussion of the Carnegie Mellon University (CMU) Software Engineering Institute (SEI) findings related to improvements to the practice of software engineering. Because software engineering affects essentially everything, it is critical that those working to ensure security and privacy be tightly engaged with those who are working to advance the state of software engineering practice.

—Deborah A. Frincke

convened a workshop on New Visions for Software Design and Productivity and Applications.[7] In their report, participants observed that, in 2001, software engineers "build vastly different systems today that are orders of magnitude more complex than those of even a decade ago."

This was possible due to prior advances—tools that support model-based code generation, application frameworks to ease web development and the design of user interfaces, and emerging software lifecycle process model improvements. Even back in 2001, the authors highlighted the importance of advancing the security and resilience of critical infrastructure, real-time and embedded systems, and distributed and mobile applications.

The community reconvened a decade later to develop a research road map titled "Future of Software Engineering Research (FoSER)."[8] Participants identified a three-step research path to move the field forward:

- understand the social context (including the development process, communication modes, and coordination norms)
- gather, analyze, and validate software process information as well as the associated artifacts (software, code reviews, bug reports, and triage records)
- analyze the gathered information.

If this is done, then (the authors claimed) we will have learned how to build larger, more complex, and interdependent systems while also being more efficient. A challenge highlighted in the report was balancing openness and privacy. The authors presciently recognized the danger of biased results due to selective data collection.

The FoSER authors also highlighted the increasing use of nonprofessionals required to act as programmers even though they do not have any formal training. These "end-user programmers" were effectively writing elementary programs with significant real-world impact, such as "specifying . . . their privacy settings, namely 'rule-based programming.'" This represented an expansion of the set of people that software engineering tools need to support and was a harbinger of today's need for more software engineers and new tools targeted to a broader range of skills.

SELECTED FINDINGS

The new National Agenda highlights findings that will resonate with computer security specialists: software engineering affects everything because software is everywhere and provides complex and critical functionality. As cybersecurity expert Dan Geer put it for our community, "cybersecurity and the future of humanity are conjoined, so it is no parlor game to figure out what capabilities we need to succeed in the future. . . ."[9] We must work together to be secure using this complex and critical functionality.

New software engineering design principles are needed, and testing them will require strategic partnerships. Today's systems are embedded in our social fabric at a global scale, bringing new challenges. Ubiquitous computing, infrastructure instrumentation, artificial intelligence (AI) advances, and distributed computing are the technical cornerstones of current and coming sociotechnical systems. The "socio" part is largely provided by human behaviors, complex social structures, culture, morality, ethics, and values.[10]

Some of these systems are here today: Facebook is now used by 35% of the world's population (up from ~10% a decade ago). These large, societal-scale systems support the desired capabilities but also have had undesirable consequences, including the rapid spread of mis- and disinformation.

Societal considerations are a new area for software engineering: it is only recently that software companies needed to address such challenging problems as: What are the appropriate limits on content when a global audience is rapidly and easily reached? What approaches to information sharing can best safeguard those desired "freedoms,"[4] e.g., by not exposing users to hidden agendas, bias, or prejudice? Proposing and designing new principles and testing implementations for intended and unintended outcomes will require access to some commercial systems to fully explore the implications.

Significant research opportunities lie at the intersection of software engineering and other fields. Connected systems and vast data as well as advances in fields like cybersecurity, AI, and machine learning are highlighted.

AI is at the heart of advances that are already affecting our lives in large and growing ways—for example, by providing assistance with tasks around the home or traveling from place to place in an autonomous vehicle. The study highlights the challenges involved in not only assuring such systems and discerning the bounds on behavior under various circumstances but also understanding emerging behaviors when such systems interact with one another.

Interestingly, however, AI is also envisioned as part of the solution, being inserted into development environments and tools in small ways today (e.g., to reduce simple programming errors). As such trends continue, the report envisions that AI-enabled capabilities may help software engineers deal with complex system interactions under rapidly changing environments.

And finally, this new study highlights widespread concern about the adequacy and availability of the software engineering workforce. Achieving all of the opportunities described in the findings without introducing new vulnerabilities that affect the software or its environment will require many exquisitely trained personnel. Studies note that the demand for highly trained personnel outstrips the supply,[11] and such shortfalls are expected to continue as the needs for software continue to increase along with the requirement for more specialized skills (e.g., in AI or in DevSecOps). New approaches will be needed to achieve the desired advances. The report envisions a workforce consisting of different skill levels, each appropriate to the task being addressed.

VISION FOR BUILDING SAFE, SECURE SOFTWARE-INTENSIVE SYSTEMS

With this backdrop and based on recent trends in research, the authors describe a vision of the future of software engineering where humans and AI are trustworthy collaborators that rapidly evolve systems based on user intent (see Figure 1). To achieve that vision and address the findings, research is recommended for both new development and architectural paradigms, each with its own research areas.

ADVANCED DEVELOPMENT PARADIGMS

Advanced development paradigms are believed to lead to efficiency and trust at scale, addressing the findings related to those topics and supporting a workforce with a wide range of skills:

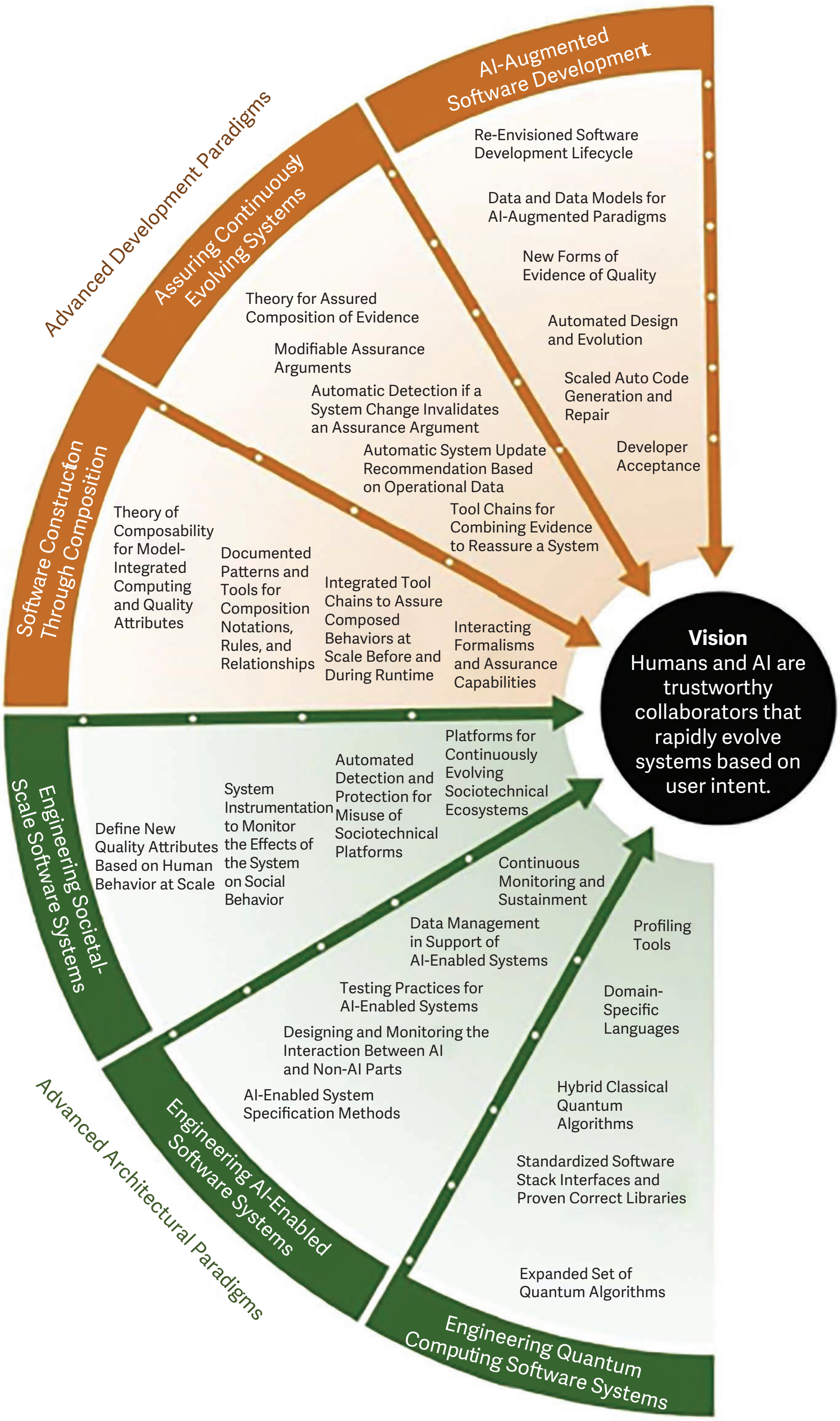


FIGURE 1. The foci for the software engineering research road map. Six areas of research are recommended, subdivided into three development and three architectural paradigms. Concrete steps are recommended for each, often starting with theoretical advances and moving toward increased practicality and complexity, supporting increasingly widespread use.

- *Humans leverage trusted AI as a workforce multiplier for all aspects of software creation:* The security and privacy (S&P) implications for this paradigm are of active interest to the S&P community, which is exploring privacy–utility tradeoffs as well as a large list of other open problems.[12]
- *Formal assurance arguments are evolved to assure and efficiently reassure continuously evolving software:* This, too, is under direct exploration and debate by the security community. Formal methods are hard to use and require thoughtful modeling by skilled practitioners. Much easier yet less secure is the "release-and-patch model." Making formal methods more accessible and rapidly applied will become essential.[9]
- *Advanced software composition mechanisms enable the predictable construction of systems at an increasingly large scale:* Security composition is a known hard problem where considerable progress has been made, yet considerable research remains to be done.[13]

All three of these paradigms are of interest to S&P researchers.

ADVANCED ARCHITECTURAL PARADIGMS

Advanced architectural paradigms are believed to enable the predictable use of new computational models, addressing findings related to the intersection of software engineering with other fields:

- *Theories and techniques drawn from the behavioral sciences are used to design large-scale sociotechnical systems, leading to more predictable outcomes:* One element of predictability will be the security of such large-scale systems.
- *AI and non-AI components interact in predictable ways to achieve enhanced mission, societal, and business goals:* As these components work together, the security community will want to ensure that as many attacks as possible can be mitigated.
- *New analysis and design methods facilitate the development of quantum-enabled systems:* Here, software, physics, and computer engineering will work together to enable a new type of computing system. For computer security personnel, changes may be forced upon us: many systems are built upon cryptography that is known to be weak if a quantum computer of sufficient size and scale can be built.

RECOMMENDATIONS

To achieve those developmental and architectural paradigms and address the findings, the study authors recommended several specific and important paths forward. These recommendations are long term (a decade or more in duration). The report provides many incremental steps to achieve each of these.[5]

1. Enable AI as a Reliable System Capability Enhancer

The software engineering and AI communities should join forces to develop a discipline of AI engineering. This should allow for the development and evolution of AI-enabled software systems that behave as intended. Moreover, it will make it possible for AI to be used as a software engineering workforce multiplier by helping with routine software engineering activities, such as generating code based on user intent, aiding in refactoring, and ensuring conformance between a system's implementation and its architecture.

2. Develop a Theory and Practice for Software Evolution and Reassurance at Scale

The software engineering research and security communities should develop a theory and associated practices for reassuring continuously evolving systems. A focal point for this research is an assurance argument or proof, which should be a software engineering artifact equal in importance to a system's architecture. Research should build upon well-known security tools, extending them so that small system changes require only incremental reassurance.

3. Develop Formal Semantics for Composition Technology

The computer science community should extend composition technology to ensure that semantics are preserved through the levels of abstraction that specify system behavior. This will allow us to reap the benefits of development by composition while achieving predictable runtime behavior.

4. Mature Engineering of Societal-Scale Sociotechnical Systems

The software engineering community should collaborate with social science communities to develop engineering principles for sociotechnical systems. Theories and techniques from disciplines such as sociology and psychology should be used to discover new design principles for sociotechnical systems, which, in turn, will result in more predictable behavior from societal-scale systems, such as social media.

5. Catalyze Increased Attention on Engineering Quantum-Enabled Software Systems

The software engineering community should collaborate with the quantum computing community to anticipate new architectural paradigms for quantum-enabled computing systems. The focus should be on understanding how the quantum computational model affects all layers of the software stack. Predictably, obtaining the benefits of quantum computing will require careful thought about abstractions and layering.

6. Prioritize Investment, Reflecting the Benefits of Software Engineering as a Critical National Capability

The software engineering community, software industry leaders, national labs, and federal departments should develop a compelling investment concept for software engineering as a national priority that increases investment in software engineering research. Industry and government resources are needed to pursue the software engineering research recommendations that enable national competitiveness with benefits across most sectors of the economy.

7. Institutionalize the Improvement of Software Engineering Advancement

Research funding organizations should work with industry and government leaders in software engineering to determine an appropriate mechanism for institutionalizing software engineering advancement through periodic reviews of the state of software engineering. This periodic review is important to track progress and maintain proficiency. This should be stipulated by law; assigned as part of the responsibility of an appropriately high level of government, such as the U.S. Office of Science and Technology Policy (https://www.whitehouse.gov/ostp/) or the U.S. President's Council of Advisors on Science and Technology (https://science.osti.gov/About/PCAST/About), and involve an appropriate cross section of the software engineering community.

8. Develop and Socialize a Strategy for Ensuring an Effective Future Workforce for the Future of Software Engineering

This problem remains challenging for the S&P community as well, with the result that tools and expertise have been stratified and some tasks outsourced to the end user (like the task of "programming" privacy settings a decade ago). The future software engineering workforce will need to have an extremely broad, interdisciplinary skill set. Since simple, experimental software can be developed with relatively little training, but mission- and safety-critical software requires many years of training and experience, the future workforce will vary considerably in depth of knowledge and experience. As this report suggests, AI will augment traditional software engineering tools, and the nature of how humans and these tools interact could change dramatically in the future.

Reviewing the history of software research over prior decades reminds us that the demands for software-enabled capabilities have only ever increased—and at a pace that has routinely challenged the current state of engineering practice. Although advances in software have emerged incrementally and organically from many sectors and enabled commercial advances that were unimaginable 20 years ago, these pieces of the software engineering puzzle do not achieve the levels of capability, safety, quality, and evolvability that future systems will require. To get there, the recommendations highlighted in this new study[5] will be needed.

The security community has pursued some research in related areas already. Composable security has been explored for several decades, and important breakthroughs have been achieved.[13] Both of our communities have a lot to do to realize that promise as well as the many others highlighted.

To achieve the vision of this study, a focused effort with continual investment and improvement in critical software engineering technologies is needed. Otherwise, assured, next-generation applications may simply not be possible.

FROM THE DEPARTMENT EDITOR

We thank Dr. Forrest J. Shull [who is also the president of the IEEE Computer Society as well as a Carnegie Mellon University Software Engineering Institute (SEI) employee] for conducting the DARPA/SEI Workshop on Software Engineering Futures that provided significant insights into the work; Anita D. Carleton for leading the study team that devised the road map; and Dr. Tom Longstaff and Dr. Robert Cunningham, along with the broader team, for their contributions during the development and documentation of the findings of the study.

ACKNOWLEDGMENTS

The authors of this article wish to acknowledge the work on the reports of many contributors and the advisory board.

This material is based upon work funded and supported by the U.S. Department of Defense under contract FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

References herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise do not necessarily constitute or imply its endorsement, recommendation, or favoring by Carnegie Mellon University or its Software Engineering Institute. Carnegie Mellon is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University (DM21-0699).

REFERENCES

1. S. Ramakrishna, "*New findings from our investigation of SunBurst*," SolarWinds. Accessed: Jan. 13, 2021. [Online]. Available: https://thwack.solarwinds.com/resources/b/community-announcements/posts/new-findings-from-our-investigation-of-sunburst
2. S. Peisert et al., "Perspectives on the SolarWinds incident," *IEEE Security Privacy*, vol. 19, no. 2, pp. 7–13, 2021. doi: 10.1109/MSEC.2021.3051235.
3. H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," *J. Econ. Persp.*, vol. 31, no. 2, pp. 211–236, 2017. doi: 10.1257/jep.31.2.211.
4. O. Tene, J. Polonetsky, and A.-R. Sadeghi, "Five freedoms for the Homo Deus," *IEEE Security Privacy*, vol. 16, no. 3, pp. 15–17, 2018. doi: 10.1109/MSP.2018.2701156.
5. A. Carleton et al., *Architecting the Future of Software Engineering: A National Agenda for Software Engineering Research and Development*. Pittsburgh, PA: Carnegie Mellon Univ. Software Engineering Inst., 2021.
6. M. Shaw, "What makes good research in software engineering?" *Int. J. Softw. Tools Technol. Transfer*, vol. 4, no. 1, pp. 1–7, 2002. doi: 10.1007/s10009-002-0083-4.
7. A. Porter and J. Sztipanovits, "*New visions for software design and productivity: Research and applications*," Nashville, TN, 2001. [Online]. Available: https://www.nitrd.gov/pubs/sdp_wrkshp_final.pdf
8. "*Future of software engineering research*," Networking and Information Technology Research and Development (NITRD) Program, 2011. [Online]. Available: https://www.nitrd.gov/pubs/FOSER_report_2011.pdf
9. D. Geer, "Auto-update considered harmful," *IEEE Security Privacy*, vol. 19, no. 2, pp. 79–80, 2021. doi: 10.1109/MSEC.2021.3050248.
10. J.-P. Steghofer, A. Diaconescu, S. Marsh, and J. Pitt, "The next generation of socio-technical systems," *IEEE Technol. Soc.*, vol. 36, no. 3, pp. 46–47, Sept. 2017. doi: 10.1109/MTS.2017.2728726.
11. T. Breaux and J. Moritz, "The 2021 software developer shortage is coming," *Commun. ACM*, vol. 64, no. 7, pp. 39–41, 2021. doi: 10.1145/3440753.
12. E. De Cristofara, "A critical overview of privacy in machine learning," *IEEE Security Privacy*, vol. 19, no. 4, pp. 19–27, July–Aug. 2021.
13. R. Canetti, "Universally composable security: A new paradigm for cryptographic protocols," in *Proc. 42nd IEEE Symp. Foundations Comput. Sci.*, Las Vegas, NV, Oct. 14–17, 2001, pp. 136–145. doi: 10.1109/SFCS.2001.959888.

ROBERT CUNNINGHAM is with the Carnegie Mellon University Software Engineering Institute, Pittsburgh, Pennsylvania, USA, and is currently the vice chancellor of research infrastructure at the University of Pittsburgh, Pittsburgh, Pennsylvania, 15260, USA. His research interests include cybersecurity, privacy, machine learning. Cunningham received a Ph.D. in cognitive and neural Systems from Boston University. He is a Fellow of IEEE. Contact him at robertkcunningham@pitt.edu.

ANITA D. CARLETON is a software solutions director at the Carnegie Mellon University Software Engineering Institute, Pittsburgh, Pennsylvania, 15213, USA. Her research interests include software engineering, software engineering measurement, software architecture, software assurance, machine learning/artificial intelligence engineering, and software acquisition practice. Carleton received a B.A. in applied mathematics from Carnegie Mellon University and an M.B.A. from the Massachusetts Institute of Technology Sloan School of Management. She is a Senior Member of IEEE. Contact her at adc@sei.cmu.edu.

TOM LONGSTAFF is a chief technology officer at the Carnegie Mellon University Software Engineering Institute, Pittsburgh, Pennsylvania, 15213, USA. His research interests include cyber, software engineering, and artificial intelligence engineering. Longstaff received a Ph.D. in computer science from the University of California, Davis. He is a Senior Member of IEEE. Contact him at tal@sei.cmu.edu.

FORREST J. SHULL is lead for Defense Software Policy Research at the Carnegie Mellon University Software Engineering Institute, Pittsburgh, Pennsylvania, 15213, USA, and is the president of the IEEE Computer Society. His research interests include empirical software engineering, software engineering measurement, and software acquisition modernization. Shull received a Ph.D. in computer science from the University of Maryland College Park. He is a Senior Member of IEEE. Contact him at fjshull@sei.cmu.edu.

EDITOR: Jeffrey Voas, j.voas@ieee.org

This article originally appeared in IT Professional vol. 24, no. 1, 2022

DEPARTMENT: CYBERSECURITY

Trust Considerations in Open Banking

Mohamad Kassab and Phil Laplante, *The Pennsylvania State University, Pennsylvania, PA, 16801, USA*

The term "open banking" defines a special kind of financial ecosystem that involves special security protocols, application interfaces, and stakeholder guidelines to provide enriched customer choices and experiences. An open banking ecosystem can provide more information and options to individuals and small to mid-size businesses. With open banking, the transfer of money and information between financial institutions and any other entity choosing to participate in the financial ecosystem is much easier than conventional banking. Open banking also opens the door to new entries into the financial business sector.[1] But because open requires exposing more user information to all participating entities, trust will be an important requirement for its success. In this article, we introduce five trust considerations needed for open banking to be successful around the world.

Open Banking (OB) is a kind of "finance as a service" (financial) Software as a Service (SaaS) where a collection of banking entities, configured as a cloud, deliver, and support various financial services via SaaS. Financial services can include deposits, withdrawals, payments, debits, credits, and origination of loans, leases, and mortgages.

An OB ecosystem allows third-party providers (TPPs) access to consumer banking, transnational, and other financial data from various institutions so long as the application programming interfaces (APIs) conform to the OB guidelines and are authorized by the customer. In OB debits and credits are executed between any products and services in the ecosystem and are executed at the discretion of the customer using conforming APIs. There are no predefined direct relationships or "supply chains" of financial products and services.[1]

OB requires strict well-structured application interfaces, security profiles, and guidelines for customer experiences and operations.[1] These guidelines are necessary in order to create the level of trust needed for customers to want to participate in an OB system.

Trust is the degree of belief that something (or someone) is reliable and safe.[2] In the context of banks; for example, most people; intrinsically, have a high degree of trust in their money and its value (though some do not, and hide their cash under the mattress). This trust enables the formation of cooperation channels among individuals and businesses. The source of trust in money comes from the trust in its issuers in preventing counterfeit. The majority of people have also a high degree of trust in banks in "managing" their money, according to a recent nCipher Security survey.[3]

1520-9202 © 2022 IEEE
Digital Object Identifier 10.1109/MITP.2021.3136015
Date of current version 15 February 2022.

NEW SERVICES IN OB

Open banking components are already deployed (partially or fully) using different approaches and names in many countries including the U.K., Europe, Mexico, Australia, Turkey, Hong Kong, Japan, Singapore, Colombia, New Zealand, and more, but not the U.S.[1] But because of the exciting possibilities, we expect a further rollout of more OB capabilities in more countries soon.

Open banking creates many new consumer services enabled by easy linking across all types of financial accounts and services from multiple, heretofore disconnected, entities. Aside from some of the more mundane examples of linking checking, savings, investment, loan, and other accounts across financial institutions (FIs), there are some exciting new possible applications of OB. For example, consider the following OB enabled use cases:

Wealth Management: With the rise in popularity of digital wealth management platforms lately, these platforms can certainly benefit from the OB system to gain a clearer context of the client before recommending the appropriate investment based on the client's payment ability and risk tolerance. Companies that can implement this use case in the U.K. include Plum (www.withplum.com), Chip (getchip.uk), and Lenlord (lendlord.io).

Buy Now Pay Later (BNPL): A small retailer wants to implement a BNPL campaign allowing users to receive their purchased items before payments are finished. A typical step in traditional BNPL programs is that customer's credit risk needs to be determined before extending credit. This step is usually outsourced by small retailers. Using an OB framework, a specialized company can smooth the interaction between retailer and customer and reduce the burden on the retailer. OB-developed apps can aggregate more information about the customer's spending habits, and using proprietary algorithms, help make a better-informed decision about the creditworthiness of a user. Companies that can implement this use case include Zilch (https://www.payzilch.com) and Klarna (https://www.klarna.com).

Improving Employee Experience: A company wants to offer its employees discount packages at retailers in their community. Typically, such a program will require the employee to present to the retailer a card proving employment at an organization that qualifies for a discount. Upon qualification, an adjustment to the retailer's point of sale system needs to be made to recognize that a discount. OB can streamline the process by connecting the employee's existing credit or debit card to their discount profile and unlocking eligible deals in their community. Moreover, augmenting the OB-developed system with AI capabilities can further enhance this benefit. By analyzing the employee's banking transactional data, the discounts can be targeted to the interests of each employee instead of a blanket discount voucher. Because there is no need to modify the vendor's system, it's also easier for a small retailer to participate in employee discount programs. Companies that can implement this use case include Perkbox (https://www.perkbox.com/uk).

Debt Collection: A customer is behind on certain loan payments. Using OB, a debt collector can look into the accounts of the person and try to generate a payment plan that the debtor can meet to pay off the remaining amount. Companies that can implement this use case include Experian (http://experian.com) and Flexys (http://flexys.com).

Carbon Tracking: An individual is interested in learning about the impact that their spending is having on the environment. An OB system connected to a carbon tracking platform can provide the user with carbon footprint insights based on their banking transactions allowing them to become more conscious about their environmental impact. The system can also offer recommendations to engage in changing the spending behaviors in a win-win ecosystem. Companies that can implement this use case include Enfuce (www.enfuce.com) and equensWorldline (https://equensworldline.com).

Because of the requirement of sharing personal and banking information between independent banking entities, these use cases will challenge the trust boundaries for many individuals.

TRUST IN OPEN BANKING

As OB is set to change the industry by introducing new services and players into the market, the pertinence of OB will depend less on the sophistication of its tools rather than the consumer perceptions (and trust) of how that OB is being used. We propose five trust considerations that need to be addressed to enhance the usefulness of OB so that customers are willing to use it.

1) Customer trust is essential for continued OB momentum, and regulations such as Payments Service Directive 2 (PSD2) and OB Implementation Entity among others are set to provide the basic foundation of trust from a regulatory perspective. For example, under PSD2, newcomers to the OB space have to receive a license from the Financial Conduct Authority (FCA) after demonstrating compliances with the data storage, business model, security, and IT practices policies, as well as possession of professional indemnity insurance. Having this approval is not an easy task, but once it is acquired, it provides evidence that a TPP has passed a high-quality bar set by the relevant authority, which signals to customers and FIs that they are to be trusted.
2) *Transparency:* The act of providing transparent and relevant information to the customer signals a company's willingness to be honest and open, and it helps to set strong trust in the business practices for OB services. To satisfy transparency in the context of OB, regulated TPPs must be ready to provide to their customers' disclosures on:
    a. the default settings for how their data are set to be shared, including what data is shared, with whom, and in what format;
    b. a mechanism to allow the customer to adjust the sharing default settings as they choose;
    c. explanation on what happens to the customers' data after the customer decides to stop

using the service with a mechanism allowing the customer to exert control over data in this context, including demanding the deletion of all data acquired;

d. explanation of how the harms of a security breach will be addressed and which parties are responsible in the various possible situations.

3) *Insurance:* The current disconnect between legal liability and practical rectification may hinder the customer's trust in OB services. For example, where data has been lawfully transmitted by FI to TPP, the responsibility for data breaches by the TPP should not position the FI in liability. But under PSD2, for example, regardless of whether an authorized transaction has occurred as a result of TPP access, the FI must still reimburse the customer, with the potential for the FI to request compensation from TPP if it is found responsible. The burden of proof can rely upon the TPP to show that it was not responsible. An industry framework to administer such compensation claims has yet to have emerged. The U.S. Bank Secrecy Act (1970) requires financial entities to report certain transaction types to the government to combat money laundering. It is unclear how that will be implemented in the OB model.

4) *Awareness:* This is naturally determinant of whether the trust should be offered to someone or something. Nevertheless, recent research showed that 52% of the Americans had no awareness of what OB was and how it can be of benefit.[4] With younger customers more open to data sharing and finding that OB is more valuable to them,[5] the level of acceptance in older generations is not as strong. And as with online banking, it will take time and effort to have everyone fully aware in OB. Clearly, the burden is on TPPs and the banks to explain what OB is to their customers.

5) *Regulation versus Agility:* While establishing regulation is critical to acquire customer trust, it may hinder the agility that OB promises. In the U.K., for example, while every TPP has to receive approval from the FCA, many TPPs are small, and not built to deal with hefty regulatory hurdles. Some TPPs have been shut down of the OB space due to the strict regulatory requirements needing to be followed to access OB infrastructure.

Despite these exciting possibilities, because it provides access to all or most of an individual's financial assets, OB will need strong cybersecurity in order to establish and maintain customers' trust.[2]

CONCLUSION

OB is emerging worldwide with well-developed guidelines and regulations and several countries have implemented workable solutions to the security and privacy problems of OB. While the U.S. has not yet developed its own OB ecosystem, many of the components needed already exist in e-banking and peer-to-peer banking services. While more implementation work is needed, experiences from other countries that are further ahead in the adoption of OB can show the way regarding cybersecurity and privacy.[6] These best practices are necessary to foster a level of consumer trust that will encourage participation.

But trust is not a static property, and it must be evaluated constantly.[2] And in a world of highly publicized hacking and ransomware attacks, OB is going to continuously challenge the trust level of its user. Therefore, building and maintaining this trust must be a focus of any entity providing OB services. We have introduced five trust considerations that must be further studied for OB to become a de facto standard way of banking.

REFERENCES

1. P. Laplante and M. Kassab, "Open banking: What it is, where it's at, and where it's going," *Computer*, vol. 2022, Jan. 2022.
2. K. Miller, J. Voas, and P. Laplante, "In trust we trust," *Computer*, vol. 43, no. 10, pp. 85–87, 2010.
3. Consumers trust banks most with their personal data, 68% still fear identity theft. Accessed: Jan. 2022. [Online]. Available: https://www.helpnetsecurity.com/2019/04/26/consumers-trust-banks-most-with-their-personal-data/
4. Axway: 52% of Americans have never heard of open banking. Accessed: Jan. 2022. [Online]. Available: https://venturebeat.com/2021/08/22/axway-48-of-americans-have-never-heard-of-open-banking/
5. V. Srinivas, J. Schoeps, and A. Jain, "Executing the open banking strategy in the United States." Accessed: Jan. 2022. [Online]. Available: https://www2.deloitte.com/us/en/insights/industry/financial-services/open-banking-model-strategy-united-states.html
6. Draft NISTIR 8389, "Considerations for open banking technology and emerging standards," U.S. Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Oct. 2021.

MOHAMAD KASSAB is currently an associate research professor in software engineering at Pennsylvania State University, State College, PA, USA. Previously, he was a postdoctoral researcher with the ETS School of Advanced Technology, Montreal, QC, Canada, and a visiting scholar at Carnegie Mellon University, Pittsburgh, PA, USA. With more than 20 years of industrial experiences, he worked in different industrial roles among which are business unit manager at Soramitsu, senior quality engineer at SAP, senior associate at Morgan Stanley, senior quality assurance specialist at NOKIA, and senior software developer at Positron Safety Systems. He has authored or coauthored articles in software engineering books and journals. His research interests include requirements engineering, system architecture, software quality and measurements, blockchain, and the Internet of Things. He received the M.S. and Ph.D. degrees in software engineering from Concordia University, Montreal. Contact him at muk36@psu.edu.

PHILLIP A. LAPLANTE is currently a professor of software and systems engineering with Pennsylvania State University, Malvern, PA, USA. His research interests include computer security, safety-critical software systems, artificial intelligence, and software quality. He received the B.S. degree in systems planning and management, M.Eng. degree in electrical engineering, and the Ph.D. degree in computer science from the Stevens Institute of Technology, Hoboken, NJ, USA, in 1983, 1986, and 1990, respectively, and the M.B.A. degree from the University of Colorado, Colorado Springs, CO, USA, in 1999. He is a fellow of the IEEE. Contact him at plaplante@psu.edu.

EDITOR: Robert Blumen, SalesForce, robert@robertblumen.com

This article originally appeared in Software vol. 38, no. 3, 2021

DEPARTMENT: SOFTWARE ENGINEERING RADIO

James Smith on Software Bugs and Quality

Priyanka Raghavan

FROM THE EDITOR

James Smith of Bugsnag discusses software bugs and quality. Host Priyanka Raghavan spoke with Smith on topics including causes, types, and history of bugs; user experience and environments causing different bugs; and measuring, bench-marking, and fixing bugs based on data. We provide summary excerpts below; to hear the full interview, visit http://www.se-radio.net or access our archives via RSS at http://feeds.feedburner.com/se-radio.—Robert Blumen

Priyanka Raghavan: Why is it okay to ship software with bugs?

James Smith: Although you should reduce bugs as much as possible before you ship, it's a tradeoff. To be competitive, you might want to deliver features or products to customers more quickly. Most importantly, you can't fully prevent bugs: you can't test every single experience customers have.

Do you see more bugs in certain languages?

Yes. JavaScript 100%. It's easy to use and is a lot of people's first language. Many junior developers pick it up and introduce bugs. It's important to understand the fundamentals of typing even in a language like JavaScript, where it's magically typed behind the scenes.

Will a particular type of architecture or design pattern have more bugs?

Digital Object Identifier 10.1109/MS.2021.3058704
Date of current version: 16 April 2021

The smaller the scope of your project and the better the contracts between your project and others, the less likely it is to have complicated, confusing bugs. This has been highlighted by the rise of microservices architectures. When an app does one thing and owns the data, you can anticipate problems that could arise more easily because you're not trying to map a complex state machine in your head. Microservices with contracts between services and applications force you to document and think about the relationship between these applications and about errors that could occur and cause the contract with other services to break down.

The contrary, interactive user interfaces, are most likely to have bugs. You're building something that people interact with in different, sometimes unanticipated, ways. Also, there's a ton of asynchronous code running. Most of your code in a UI, web, desktop, or mobile app is running in callbacks, waiting for someone to interact with your application. An exception in a callback doesn't kill execution for the rest of the application, it just causes that callback to fail. So for the customer, the whole application keeps working, but just your callback or your click handler might break.

How do you handle bugs coming from third-party libraries?

Fortunately, most people are using open source third-party libraries these days. You shouldn't have bugs in third-party code, but you will. If it's open source, at least you can go and dig into it. If you're using an error-reporting, error-monitoring solution, it will show you the stack trace, the line of code that caused the crash, and all of the other code paths that the customer went through before the crash.

You shouldn't live on the bleeding edge. It's exciting to get hot new features, but you shouldn't immediately bump your dependencies as soon as something new comes out in beta. Selection of third-party libraries is an underrated part of software development. If you rely on something, you need to trust it, so researching third-party libraries and SDKs is critical and underrated.

Can some classes of bugs be found only by actual users in the field?

Huge teams used to work for months on QA going through QA scripts. The more we've gotten to lean, agile, rapid iteration, and being able to hot fix and patch things and componentize software, the faster we can ship. You can't now have a team of humans do two months of QA.

The left-hand side of software development has been replaced with what Capital One calls "team quality engineering"—trying to automate that as much as possible. From the right-hand side, you have data-driven instrumentation, with products that will tell you, "This is a problem, this is how many customers it affected, and this is how you fix it."

Bugs exist in the hands of customers, where data representing that user has gotten into a strange state. Preproduction and precustomer testing include unit and integration tests and cleanroom environments. But in reality, customers run software in a dirty environment because it needs to do things such as save state, cache, and authenticate the customer.

Most bugs are not due to code paths being missed, because there is usually good code coverage in testing. Most are about weird data structures and unclean data coming through. The problems that happen in the hands of your customers come from caching, authentication, cookies, local storage, and stuff that's stored on the device that is not in the format you expect.

Is it okay to delegate fighting bugs to our clients who paid money for software?

I think "test in production" is the wrong way of thinking about things. You have to be intentional about tradeoffs. We want to use tooling and technology to remediate bugs as quickly as possible. If resources are scarce, as they always are, fix the bugs that matter the most. That will vary by company and product.

You care most about bugs that affect key customers or that are happening in an important flow, such as login or payments. If it's a consumer mobile application that doesn't have people spending a lot of

SOFTWARE ENGINEERING RADIO

Visit www.se-radio.net to listen to these and other insightful hour-long podcasts.

RECENT EPISODES

- 446—Nigel Poulton, author of *The Kubernetes Book* and *Docker Deep Dive*, discusses Kubernetes fundamentals, why Kubernetes is gaining so much momentum, and why it's considered "the" Cloud OS with host Gavin Henry.
- 445—Host Justin Beyer spoke with Thomas Graf, cofounder of Cilium, to discuss eBPF and how it can be leveraged to improve kernel-level visibility.
- 444—Host Akshay and Tug Grall of Redis Labs discusses Redis, its evolution over the years, and the emerging use cases today.

UPCOMING EPISODES

- Host Adam Conrad talks with Dan Moore about build versus buy.
- Luke Kysow discusses service mesh with host Priyanka Raghavan.
- Host Jeremy Jung talks with Scott Hanselman about the Microsoft .NET framework.

money, focus your time on bugs affecting the highest volume of customers. Whatever metric you use, be thoughtful about it, and use data to drive it. Then you can deliver software that's as stable as if you did a two-month QA process, and in fact improve it and get features to market more quickly.

There are great tools out there that support a data-driven approach to prioritizing and fixing bugs. But even if you do this yourself, don't wait for customers to complain. By the time customers have complained, probably 50 other customers have already abandoned your product.

How do you use a stability score?

We want to understand what percentage of user interactions with your product are good or have failed. We will detect if there are unhandled exceptions, unhandled promise rejections, or exceptions in a callback. Or you may be using a framework that detects errors that cross an error boundary. So we build these hooks to failure states in your product.

These will cause your customers to have a bad experience. We don't magically stop bugs from happening, but we detect when they do happen. That all then feeds into the stability score. If one of those scenarios happens in your session, that counts as a failed session. The underlying concept is, I want to know which customers, what percentage of the customer base, had a positive experience.

PRIYANKA RAGHAVAN is a security architect at Maersk, Bangalore, 560077, India, where she makes day-to-day architectural decisions between security and usability or performance. She has more than 15 years in the software industry playing various roles from developer, team lead, software architect, and now security architect across three continents. She can be reached on Twitter @Priyankarags. Further information about her can be found at https://www.linkedin.com/in/priyanka-raghavan-6a1a405/. Contact her at priyankaraghavan@gmail.com.

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: Direct unresolved complaints to 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.

AVAILABLE INFORMATION: To check membership status, report an address change, or obtain more information on any of the following, email Customer Service at help@computer.org or call +1 714 821 8380 (international) or our toll-free number, +1 800 272 6657 (US):

- Membership applications
- Publications catalog
- Draft standards and order forms
- Technical committee list
- Technical committee application
- Chapter start-up procedures
- Student scholarship information
- Volunteer leaders/staff directory
- IEEE senior member grade application (requires 10 years practice and significant performance in five of those 10)

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 Society publishes 12 magazines, 19 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 society holds more than 215 conferences each year and sponsors many educational activities, including computing science accreditation.

Certifications: The society offers three software developer credentials.

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

MEMBERSHIP & PUBLICATION ORDERS
Phone: +1 800 272 6657; **Fax:** +1 714 821 4641;
Email: help@computer.org

IEEE

revised 6 February 2023

EDITORS: Mary Ann Leung, mleung@shinstitute.org
Ann Jeffers, jffrs@umich.edu

This article originally appeared in computing in SCIENCE & ENGINEERING vol. 23, no. 6, 2021

DEPARTMENT: DIVERSITY & INCLUSION

Inclusivity Bugs and the Language We Use

Mark C. Miller, Lawrence Livermore National Laboratory, Livermore, CA, 94550, USA

If we knew many terms and phrases we commonly use such as, for example, *sanity check*, *grandfathered*, *ladies and gentlemen* or *low man on the totem pole* can discourage coworkers, colleagues, collaborators and other potential contributors from participating in our projects with their full authentic selves, would we still use them?

When Armstrong first set foot upon the Moon, he said, "This is a small step for a man but a giant leap for mankind." Minutes later, responding to President Nixon, he said, "It is a great honor. . .to be here representing. . .men of peaceable nations, men with an interest and a curiosity, and men with a vision for the future."[a]

Taken out context of the 1960s, the repeated use of man and men certainly doesn't read very much like space exploration included women. More importantly, however, it also didn't sound that way to many of the ~650 million listeners at the time.[b] Worse, even *current* writings about the Apollo space program tend to parrot the same language using phrases like *when man first walked on the Moon* or what a *great achievement it was for mankind*.

To help skeptics empathize, imagine if instead of a crew of all white, Christian men as the first Apollo mission was (maybe all Apollo missions), the first mission to Mars is crewed by all Black, Muslim women, and the first to walk on Mars says something like *"This is a small step for a woman but a giant leap for womankind. . .Allahu Akbar."*

Joanna Lee, a contributor to the Inclusive Naming Initiative and author of inclusive language guidelines on inclusiveprism.org, calls the use of such language in software projects an *inclusivity bug* because it creates barriers to full participation by all stakeholders.[c–e] Inclusivity bugs arise out of the use of language that discourages whole groups of people, often from underrepresented and/or historically oppressed populations, from contributing as their full authentic selves. This can happen regardless of how common-place any manner of speaking or turn of phrase is or may have been for its time. In fact, an alarming variety of inclusivity bugs stem from language we all commonly use.

Readers might dismiss Armstrong's specific words as merely a figure of speech or a sign of the times. *That's just the way we spoke then. Armstrong didn't intend to exclude women.* Or, readers might think that the problem is the English language because it does not easily avail itself to gender neutrality. But, that would be wrong. While English is notoriously lacking in gender-neutral pronouns, it is otherwise one of the most accommodating languages for gender neutrality.

As HPC/CSE software developers maybe we're not doing anything nearly as momentous as being the first Earthlings to walk on another planet. Do we really need to worry so much about inclusive language? If we care about our project's reach and maximizing our ability to attract and retain collaborators, developers, users, or sponsors, then yes we do. That is because the language we use has the power to welcome others in as well as push them away.

Software projects involve a lot of communication. As developers, we write code, comments, documentation, research papers, presentations, job postings, promotional materials, discussion posts, emails, chats, tweets, and more. It is now well known that language used in job postings has the effect of biasing the respondent

[a]https://www.presidency.ucsb.edu/documents/telephone-conversation-with-the-apollo-11-astronauts-the-moon
[b]https://en.wikipedia.org/w/index.php?title=Apollo_11_in_popular_culture

[c]https://www.gesmer.com/team/joanna-lee/
[d]https://inclusivenaming.org/
[e]http://www.inclusiveprism.org/

1521-9615 © 2021 IEEE
Digital Object Identifier 10.1109/MCSE.2021.3125473
Date of current version 16 December 2021.

pool.[f] The more we adopt inclusive language, the more we reduce biases and create welcoming and safe spaces for all participants to be their authentic selves, to be productive, and to thrive.

Any readers who have written proposals likely already appreciate the role language plays when communicating our work. When writing proposals, we often struggle with and debate at length how individual word choices will align our proposal with the sponsor's call. In other words, we all already appreciate the importance of and can be quite mindful of the impact of language on funding. Being mindful of the impact of language on inclusion is no different.

INCLUSIVE NAMING *IS CLOSELY RELATED TO INCLUSIVE LANGUAGE BUT FOCUSES PRIMARILY ON THE NAMING OF THINGS.*

Inclusive *naming* is closely related to inclusive language but focuses primarily on the naming of things. That is the *names* we choose for abstract objects of our software systems. A special challenge with inclusive *naming* is that people tend to get pretty attached to names. Was Edmund Halley the first to observe and record the comet that bears his name? Chinese astronomers observed and recorded it almost 2000 years earlier! Was Pythagoras the first person ever to discover and write down the rule we know as the Pythagorean theorem? Plenty of scholarly research says others developed it earlier.[g–k]

For many terms and phrases in common use, we are simply unwittingly parroting others. That can be problematic when one is not mindful of the meaning, history, or impact of a term. For example, why do we call defects in software *bugs*? How might women feel reading documentation that constantly refers to the user as *he*, *his,* or *him* ? How might Black people feel reading that existing users will be *grandfathered in* when the software license terms are changed? How might neurodiverse people feel about being tasked to add *sanity checks* to a software package's test suite?[l–o]

By the way, do readers recognize any bias in the preceding questions? It's subtle but worth mentioning. The way these questions are worded assume that only women would care about documentation that is laden with male pronouns or that only Black people would care about language derived out of racially oppressive practices or that only neurodiverse people would care about language associated with atypical patterns of thought or behavior. The truth is, we all (should) care. And, to use language that assumes or suggests otherwise is, well, not inclusive.

No way to phrase something will be acceptable to literally *every* individual. And, there is no reason we should be aiming for that either. Catering to any *one* individual's tastes and sensibilities is not what inclusive language is all about. The goal of inclusive language is to reduce the use of terms and phrases that discourage whole groups of people from participating as their full authentic selves. It is not about being politically correct or being the language police. It is not about avoiding offending people either.[p] It's about being willing to acknowledge that certain terms and phrases (and, honestly, even names, icons, and logos), however common in current culture, can be unnecessarily exclusive and being willing to consider and adopt alternatives that are less so.

If you've read this far, you may be asking yourself, how do I get started? We suggest reading some of the resources available from other major organizations such as the federal government's *Plain Language*, Google's *Inclusive Style*, or Microsoft's *Bias Free Communication.*[q–s] In particular, if you are looking for *tooling* to help alert you to inclusive language issues, some

[f]https://www.mya.com/blog/unconscious-bias-in-job-descriptions/
[g]https://www.space.com/19878-halleys-comet.html
[h]The comet is named for Halley not because he saw or recorded it first, but because he is believed to be the first person to have estimated its orbit and based on that connected recorded observations from 1531, 1607, and 1682 as being the same celestial object from which he correctly predicted its return in 1758. That said, apart from the very significant achievement of the comet's orbital estimation, it isn't clear if astronomers much earlier than Halley had recognized this repeat visitor as the same celestial object.
[i]https://en.wikipedia.org/wiki/Historical_comet_observations_in_China#Halley's_Comet
[j]https://en.wikipedia.org/wiki/Pythagoras#In_mathematics
[k]https://www.researchgate.net/publication/337941217_Mathematics_in_Ancient_Egypt_Part_II
[l]https://en.wikipedia.org/wiki/Software_bug#History
[m]https://www.washingtonpost.com/world/2019/12/15/guide-how-gender-neutral-language-is-developing-around-world/
[n]https://www.npr.org/sections/codeswitch/2013/10/21/239081586/the-racial-history-of-the-grandfather-clause
[o]https://gist.github.com/seanmhanson/fe370c2d8bd2b3228680e38899baf5cc
[p]https://theconversation.com/why-people-take-offence-131736
[q]https://www.plainlanguage.gov/
[r]https://developers.google.com/style/inclusive-documentation
[s]https://docs.microsoft.com/en-us/style-guide/bias-free-communication

resources provide tooling as a web service. That said, community standards and DevOps-hardened tooling for inclusive language is similar to spelling or grammar checkers and still in its infancy stages.

Ever since adopting the practice myself, not a week goes by that I don't have the experience of questioning a term or phrase I am about to use. I often spend a few minutes searching the web to learn more about it. This includes terms and phrases I have used many times before without really thinking about it such as describing an off-topic enhancement request as having gone *off the reservation*, or replying to it with *no can do*, or introducing a colleague as a Python *Guru* or a Java *Ninja*, or complaining my bug reports are *falling on deaf ears*, or that the level of effort is too many *man-hours*.[t–x]

WHEN YOU ARE SEEKING TO LEARN MORE ABOUT ANY POSSIBLE NEGATIVE IMPACTS OF A GIVEN WORD OR PHRASE, IT'S BEST TO SEEK OPINIONS FROM EXPERTS WITHIN *THE GROUPS MOST LIKELY IMPACTED.*

Many online resources may be useful to readers who would be similarly inclined. But, beware. In seeking greater understanding of the history of terms and phrases you thought you knew, you find a lot of misinformation out there to sift through.[y] In addition, when you are seeking to learn more about any possible negative impacts of a given word or phrase, it's best to seek opinions from experts *within* the groups most likely impacted. In other words, if a given phrase has potential negative impacts against Blacks, then language experts within the Black community will likely have the most authoritative guidance. If a given phrase has potential negative impacts against the deaf, then language experts within the deaf community will likely have the most informed guidance.

[t]https://www.npr.org/sections/codeswitch/2014/06/29/326690947/should-saying-someone-is-off-the-reservation-be-off-limits
[u]https://mentalfloss.com/article/625916/racist-origins-common-phrases
[v]https://www.yogajournal.com/lifestyle/cultural-appropriation/
[w]https://www.hearinglikeme.com/is-the-phrase-falling-on-deaf-ears-offensive/
[x]https://bossbetty.com/big-story/manpower-man-hours-and-other-phrases-to-ix-nay-from-your-iased-bay-vocabulary/
[y]https://www.nytimes.com/1998/01/25/magazine/on-language-misrule-of-thumb.html

Finally, in seeking to fix inclusivity bugs, it's important to take care that we don't introduce another kind of problem. . .excluding the use of perfectly acceptable language for no other reason than out of fear of looking bad or being labeled noninclusive. A good example is the word *master* alone, wholly apart from *slave*.[z] Hopefully, we all can agree that *master/slave* language is not acceptable.[aa]

But, there is much less agreement, notably even within the Black community, about the word *master* alone and with no relation to *slave*. Many uses of *master* have no historical roots in oppressive or genocidal systems. These include such terms as *mastermind*, *postmaster*, *master key*, *master recording*, and achieving *mastery* of a skill, and even common tech terms like *webmaster* and *scrum master*. This issue garnered much attention when GitHub announced it would change its *default* branch name (which most users simply adopt without question) from *master* to *main*.[bb] In that move, it is worth pointing out that the *default* language of a widely used resource such as git is a significantly greater proliferation potential and represents a qualitatively different situation than the choices individual projects make. Furthermore, no project is prevented from using *master* as a git branch name if they choose.

Given the current social justice climate in which we all operate, some readers will feel that inclusive language efforts are nowhere near enough to meet the moment and are really just a distraction from bigger issues.[aa] It's hard to argue with that. Nonetheless, others will feel like such efforts go way too far. While we can acknowledge both perspectives exist, it is worth considering that when we're accustomed to parroting the status quo, pausing to be more inclusive may feel like oppression.[cc]

MARK C. MILLER is currently a Computer Scientist supporting the Weapon Simulation and Computing (WSC) program at Lawrence Livermore National Laboratory for more than 30 years. He has recently developed a passion for addressing issues at the nexus of computing and inclusion. Among other things, Mark contributes to VisIt, Silo, HDF5, and IDEAS-ECP. Contact him at miller86@llnl.gov.

[z]https://www.etymonline.com/word/master
[aa]https://www.wired.com/story/tech-confronts-use-labels-master-slave/
[bb]https://www.vice.com/en/article/k7qbyv/github-to-remove-masterslave-terminology-from-its-platform
[cc]https://www.huffpost.com/entry/when-youre-accustomed-to-privilege_b_9460662

This article originally appeared in pervasive COMPUTING vol. 21, no. 2, 2022

Smart Homes or Real Homes: Building a Smarter Grid With "Dumb" Houses

Sean Barker, Bowdoin College, Brunswick, ME, 04011, USA

Dylan Parsons, Bloomberg LP, New York, NY, 10022, USA

Smart homes present numerous potential advantages for home occupants and utilities, including benefits such as energy savings, user convenience, and detailed data collection. Realizing any of these benefits requires a level of home instrumentation, such as the deployment of sensors or upgraded devices within the home. However, due to the expense and overhead of installing such equipment, instrumentation happens slowly and gradually, leading to many homes that are not particularly "smart," yet should still function as part of the emerging smart grid. As a result, real-world smart home applications, particularly those that operate across many homes, will need to be realized in homes with modest and nonstandardized smart home capabilities. As a case study, we consider the problem of monitoring the energy use of specific devices across many homes without smart home capabilities and describe a prototype system to tackle this problem. Considering such systems may broaden our notions of what constitutes a "smart home" at all.

WHAT MAKES A SMART HOME?

A smart home, by its very nature, requires some level of built-in intelligence, whether through automated data collection (e.g., smart energy meters, motion sensors, network-connected cameras, etc.) and/or devices allowing automated control of the home environment (e.g., smart thermostats, programmable light switches, network-connected appliances, etc.). The traditional notion of a smart home often includes a significant amount of both—e.g., a home in which heating and cooling are controlled automatically, lighting and major appliances can be operated based on a schedule, energy use is optimized to reduce waste and minimize costs, etc. In practice, real-world homes rarely approach this idealized picture. Apart from major retrofits or new, state-of-the-art homes built with these goals in mind, most homes that make use of smart home capabilities are mostly "dumb," but have been enhanced with a small number of "smart" capabilities. For example, a home might have a smart energy meter providing the homeowner with data on their energy use, whereas another home might have a smart thermostat automating their heating schedule, and another home might have lights that occupants can control from a computer or a phone app. In short, most homes are neither completely "smart" nor completely "dumb," but rather make small, *incremental* movements toward intelligence. Should such homes be considered "smart homes"? Whether yes or no, many smart homes of the future will emerge in this way—evolving through slow, gradual changes rather than sudden, all-encompassing upgrades. Commercial smart home products accommodate this evolution by supporting a mix-and-match of multiple smart home devices performing various functions, and companies, such as Apple, Amazon, Samsung, and Google, all offer smart home hubs that are designed for expansion as future smart home products are installed. A typical real-world smart home will contain some subset of these products, and no two homes are likely to be exactly alike in their capabilities.

INCREMENTALLY SMART GRID

Visions of next-generation smart grids typically assume homes with extensive instrumentation and advanced data-collection and control capabilities. In principle, the smart grid can leverage the capabilities of such homes to coordinate at the scale of a neighborhood or city (e.g., to stabilize energy consumption by flattening the demand on the electric grid). In practice, however, since most homes only have limited and highly specific functionality, coordination of these capabilities may be difficult. Instead, it may be more practical to envision smart home and smart grid applications based on a minimum level of easily and cheaply realized functionality, without

1536-1268 © 2022 IEEE
Digital Object Identifier 10.1109/MPRV.2022.3160752
Date of current version 28 June 2022.

worrying about whether such functionality is or is not already present in homes.

As an example, consider the specific challenge of distributed energy monitoring—e.g., tracking the energy consumption of air conditioners across many homes. Within any specific home, energy data might be gathered from a variety of sources—a smart utility meter providing aggregate consumption readings, a submetered electrical panel providing circuit-level readings, smart energy plugs providing device-level readings, etc. Homes may have all, some, or none of these functions, and these functions might be provided by many different types of equipment. As a result, distributed data collection (such as by an electrical utility wishing to monitor particular kinds of devices) is nontrivial at scale; for example, large-scale sensing in homes tends to result in connectivity, maintenance, and user-related challenges.[1] One simple way to manage these types of challenges is by targeting the minimum level of smart home functionality needed to implement the desired application—i.e., without assuming or implementing any further level of "smarts" within the homes involved.

WATTCHER: A SELF-CONTAINED "SMART" ENERGY MONITORING SYSTEM

As a case study of such an approach, we implemented a system for distributed energy monitoring across large numbers of homes that assumes no existing functionality (hubs, sensors, meters, computers, etc.) except for an accessible Wi-Fi network in each home. Our system, called WATTcher, supports rapid and inexpensive deployment across many homes and is designed to be installed unassisted by nontechnical users.

The building block of WATTcher is what we term a *microdeployment*, which contains one or more energy sensors attached to devices of interest. Rather than aiming to collect data from the entire house through extensive instrumentation, a WATTcher microdeployment is intended to be small, lightweight, and easily deployed—perhaps only monitoring one or two devices of interest in a home. Figure 1 visualizes the WATTcher system architecture, which contains multiple active microdeployments coupled with a centralized data repository that provides long-term storage and querying capabilities.

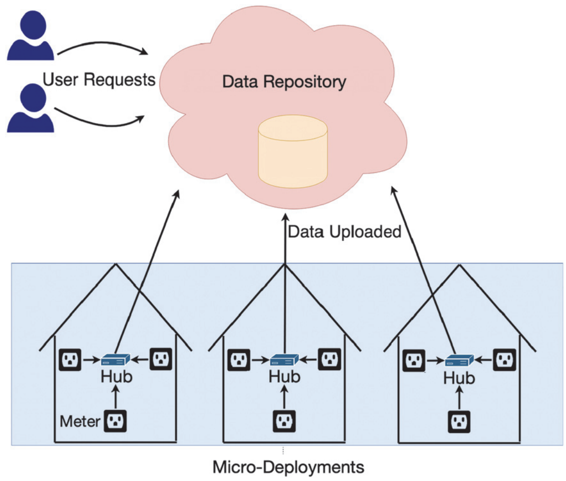


FIGURE 1. Architecture of the WATTcher sensing system.

MICRODEPLOYMENTS

Each microdeployment consists of at least two components: a plug-level energy meter and a collection hub that stores data readings from meters and periodically transmits them via a local Wi-Fi network to the centralized data repository. While a microdeployment may consist of a hub and only a single meter, additional meters can be added to expand any given deployment.

Given the meter and hub hardware, installing a microdeployment in a new home takes only a few minutes, even for a non-technical user. Specifically, installation requires the following three simple steps (the third of which is optional).

1) Connecting the collection hub and energy meter to power sources.
2) Providing Wi-Fi credentials to the collection hub via a web page (hosted on the hub itself).
3) Optionally, labeling the metered device(s) (e.g., "television").

To receive configuration information from the user, the collection hub initially acts as a Wi-Fi access point and advertises a public network upon being powered ON. After connecting to the hub network from a phone or computer and opening a browser, the user is presented with a page requesting local network credentials and optionally a device name. After submitting this information, the hub switches from an access point to a Wi-Fi client and connects to the local network, then begins transmitting data collected from the local energy meter to the central repository. At this point, the local microdeployment becomes part of the global WATTcher system and is centrally managed without any user involvement as long as network connectivity is maintained.

The centralized data repository stores data uploaded by microdeployments, tracks and manages the microdeployments themselves (e.g., pushing software updates

TABLE 1. Approximate Component Costs of a Single-Device Microdeployment.

Component	Cost (USD)
Aeotec Z-Stick	$45
Monoprice Z-Wave Energy Meter	$20
Raspberry Pi Zero W	$10
32-GB SanDisk microSD card	$8
5-V 2.5-A power cord	$7
MicroUSB to USB hub	$6
Simco hard shell case	$4
Total Cost	**$100**

FIGURE 2. Internal components of a WATTcher hub (power cable on right).

and tracking disconnected microdeployments), and provides an interface for users to view and manage their data (e.g., relabeling devices or adding additional meters).

HARDWARE AND SOFTWARE COMPONENTS

WATTcher is built using inexpensive, off-the-shelf parts. The hub is a Raspberry Pi Zero W, a single board computer that consumes less than 2 W of power and retails for only $10 USD. The Pi provides built-in Wi-Fi and USB connectivity in a package the size of a credit card, making it an ideal candidate for an unobtrusive hub.

Hubs and meters communicate using the Z-Wave protocol, which provides wireless operation and a sufficient bitrate for typical data resolutions (e.g., 1 Hz). There are many off-the-shelf Z-Wave energy meters available; we use a Monoprice model that occupies only about as much space as an outlet itself. A USB Z-Wave adapter (Z-Stick) attached to the hub enables communication with the Z-Wave energy meters. A few final components are necessary to complete a microdeployment: a microSD card for system and local data storage, a power cable for the hub, and a micro USB to a USB hub that is necessary to connect the Z-Stick to the Pi. Finally, to ensure that the hub is neither conspicuous nor prone to damage, we enclose all hub components within a hard plastic case through which the only opening is a drilled hole for a power cable.

Table 1 gives a listing of the hardware components and their approximate costs. The total cost of a single microdeployment is roughly $100 (of which nearly half is for the Z-Stick alone). Expanding a microdeployment only requires adding additional meters, at a cost of $20 per meter with our current hardware.

The internal components of the hub are shown in Figure 2, whereas a final assembled hub is shown in Figure 3. An active energy meter transmitting to a hub is shown in Figure 4.

Programs running on the hub gather data from the meters, periodically connect to the central repository to upload data, and manage switches between access point mode (for user configuration) and client mode (for regular operation). We presently collect data at 1 Hz, although the current meters support higher resolutions as well. Figure 5 shows an example of historical 1-Hz data recorded from a television in our deployment.

The central repository is designed to facilitate easy communication with individual microdeployments without requiring any special network access within homes. For example, home networks often employ network address translation, dynamically assigned IP

FIGURE 3. Hub enclosure (left) containing the components pictured in Figure 2. Z-Wave energy meter (right) shown for size reference.

FIGURE 4. Energy meter connected to a WATTcher hub monitoring a coffeepot.

addresses, and/or restrictive firewalls, all of which complicate any communications initiated by the central repository. Rather than placing any burden on users to reconfigure their network, the repository operates using a polling-based model initiated by the client hubs in each microdeployment. Each hub stores data readings locally, then periodically connects to the central repository to upload recent data and check for administrative instructions (e.g., software updates to apply to the hub). In the event that a hub is repeatedly unable to contact the repository (e.g., due to a network outage or changed local Wi-Fi credentials), the hub automatically switches back into setup mode, allowing it to be reconfigured by a local user before it reconnects to the repository. The primary advantage of this communication model is that there are no requirements placed on individual home networks; no inbound connections to the hub are ever initiated by the repository or any other machine not on the local network.

We have assembled roughly 20 hubs and deployed them in and around a New England college campus (offices, dormitories, public spaces, and residential houses). The current WATTcher deployment gathers close to 10 million data points each month. We envision using a system such as WATTcher to collect data on devices of interest that may not even be present in most houses—e.g., car chargers, exercise equipment, specific large appliances, etc. Such data would be complementary to the many existing public energy datasets (e.g., the datasets by Kelly and Knottenbelt[2] and Kolter and Johnson[3]) that focus on extensive instrumentation of a small number of homes, rather than specific devices spread over large numbers of homes (with only a few monitored devices in each home).

REIMAGINING THE "SMART HOME"

A home containing a WATTcher microdeployment (or any similar system) is not particularly sophisticated; in our case, it may contain nothing more than a few cheap devices operating in isolation from the rest of the (conventional) home infrastructure. One might reasonably claim that such a home is not, in fact, a smart home; yet, such homes can still be used to realize useful smart home and smart grid applications. Moreover, the fact that most homes have yet to realize sophisticated smart home capabilities is a reason to focus on these kinds of applications, which build on only the most common types of in-home capabilities (e.g., an ordinary Wi-Fi network) as opposed to more

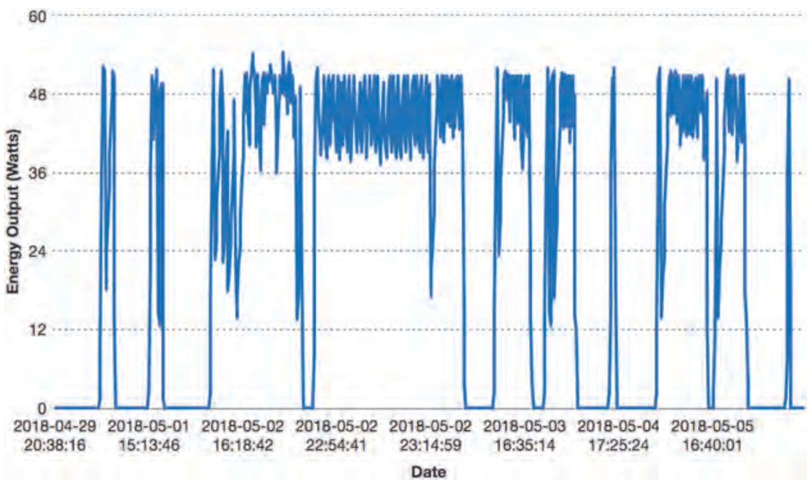


FIGURE 5. Week of data from a microdeployment monitoring a television.

advanced but less common capabilities (e.g., a high-resolution smart energy meter or smart outlets). These sorts of applications may be more modest in their capabilities, but can be more readily realized to benefit many users—whether or not these users are living in homes that are particularly "smart" to begin with.

REFERENCES

1. T. Hnat *et al.*, "The hitchhiker's guide to successful residential sensing deployments," in *Proc. 9th ACM Conf. Embedded Networked Sensor Syst.*, Seattle, WA, USA, Nov. 2011, pp. 232–245.
2. J. Kelly and W. Knottenbelt, "The U.K.-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five U.K. homes," *Sci. Data*, vol. 2, 2015, Art. no. 150007, doi: 10.1038/sdata.2015.7.
3. J. Zico Kolter and M. J. Johnson, "REDD: A public data set for energy disaggregation research," in *Proc. 1st KDD Workshop Data Mining Appl. Sustainability*, San Diego, CA, USA, Aug. 2011.

SEAN BARKER is currently an assistant professor of computer science at Bowdoin College, Brunswick, ME, 04011, USA. Barker received his Ph.D. degree in computer science from the University of Massachusetts Amherst. He is the corresponding author of this article. Contact him at sbarker@bowdoin.edu.

DYLAN PARSONS is currently a software engineer at Bloomberg LP, New York, NY, 10022, USA. Parsons received his B.A. degree in computer science and mathematics from Bowdoin College. Contact him at dylan.parsons1@gmail.com.

EDITOR: Joanna F. DeFranco, Pennsylvania State University, jfd104@psu.edu

DEPARTMENT: INTERNET OF THINGS

This article originally appeared in Computer vol. 55, no. 2, 2022

Alexa, M.D.

Sarah Jane Mee and Salam Daher, *New Jersey Institute of Technology*

In the health-care field, intelligent personal assistants (IPAs) are commonly used as diagnostic guides for health-care professionals or support for patients. This article expands the use of IPAs in health care by demonstrating using Alexa to train health-care personnel.

The Internet of Things (IoT) is at the forefront of new developments in software and technology. An intelligent personal assistant (IPA) is a type of IoT device that can respond to voice commands, complete basic tasks, and mimic the structure of human conversation. Popular IPAs include Amazon's Alexa, Apple's Siri, Google's Google Assistant, and Microsoft's Cortana. These services are used for daily tasks such as getting directions and making grocery lists as well as more complex tasks in health care and business. Within the health-care field, IPAs are often used as knowledge databases or diagnostic guides for health-care professionals and patients. For example, university researchers developed an Alexa skill to deliver interventions to metastatic breast cancer patients.[10] Another organization created an Alexa skill to help patients inquire about their own health-care plan and prescriptions.[11]

In this article, we explore the technical capabilities of Amazon's Alexa to create a customizable training model for health-care training. Human conversation is needed to train health-care providers (e.g., doctors, nurses) in scenarios such as a review of systems where they ask the patient questions about body systems to obtain his or her medical history and current health status. We developed an Alexa skill that can respond to user prompts using dynamic content provided by a subject-matter expert, initiate actions/questions, and provide custom feedback. We combined Alexa with dynamic content using the Google Sheets and Google Drive application programming interfaces (APIs), offered through the Google Cloud Platform. We describe the development process and present the technology's capabilities, limitations, and tips for optimization. This software opens the door to future research into the effectiveness and practicality of using this technology as a training tool for health-care students and professionals.

Digital Object Identifier 10.1109/MC.2021.3126528
Date of current version: 14 February 2022

IPAS

Voice-driven communication with devices is a common feature in the current world of technology. IPAs such as Amazon's Alexa, Apple's Siri, Google's Google Assistant, and Microsoft's Cortana allow people to communicate with their devices in a hands-free manner and complete tasks such as getting directions, searching for recipes, setting reminders, messaging, checking the weather, and controlling smart devices. These intelligent devices are at the leading edge of technology advancements in artificial intelligence, speech recognition, and natural language processing and are being widely implemented in fields such as business, health care, and education. Although still relatively new, voice-focused IPAs provide more flexibility and applicability than the digital assistants of the early 2000s.

Uses of IPAs by the general public

The Pew Research Center (2020) reported that 46% of American adults use digital assistants, with 42% of the public using them on smartphones, 14% on a computer or tablet, and 8% on a standalone device as their medium.[1] Three categories of the most common voice commands are music, hands-free searching, and IoT control. IPA usage for IoT control involves manipulating devices such as smart thermostats, lights, speakers, and cameras. Other popular command categories include volume control, weather reports, timers, and alarms.[2]

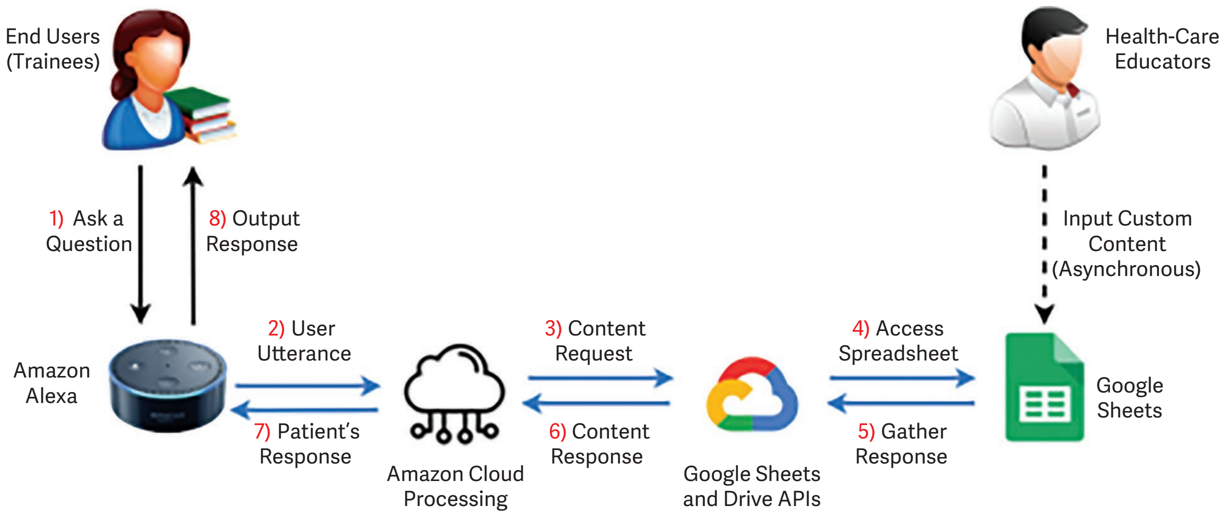


FIGURE 1. An overview of skill execution. API: application programming interface.

IPA usage in health care

Although growing, the usage of IPAs in the health-care field is much less than in other areas like business and education.[3–4] Within the health-care field, IPAs can be used to help patients follow a medical routine, track eating and sleeping habits, and monitor signals such as heart rate and blood pressure.[4] General users tend to ask IPAs about illness symptoms or medication information, while broader uses in health care involve aiding users in scheduling doctors' appointments, setting and executing reminders for medication, and maintaining a health schedule.[5]

Customizable IPAs

IPAs come with a wide range of built-in skills, but they are often inflexible and difficult to customize.[6] Customizable content is extremely important for training health-care providers. There is literature and anecdotal evidence from health-care educators that there is a need for flexible and customizable IPAs with skills to assist users in creating personalized content.[1–6]

We created a review of systems' Alexa skill, where the user can ask Alexa (that is, the patient) a series of questions to obtain past and current health information. We combined Alexa and Google APIs to create custom content for Alexa to add variability to Alexa's responses, keep score of users' questions, have time-triggered events during skill execution, and provide custom performance feedback.

METHODOLOGY

We used Amazon Echo Show 8 (First Generation, 2019 Release) to gain access to the Alexa software, but any device with Alexa capabilities can be used. An Amazon Developer Account grants access to the Alexa Skills Kit and Developer Console,[7] which is needed to build a custom, Alexa-hosted Python skill and interface with Google Sheets. We used the Google Cloud Platform to access the Google Developer Console[8] and enabled the Google Drive and Google Sheets APIs on our project. Finally, we imported the gspread library[9] to provide functionality for opening, reading from, and writing to Google spreadsheets (see Figure 1).

Custom content for Alexa

Each question that a user can ask Alexa is an intent, and the response to an intent is managed by an intent handler in the back end. We edited the content of the intents in the front end and changed Alexa's responses in the back end to customize the questions and answers that Alexa can interpret in this skill.

AN INTELLIGENT PERSONAL ASSISTANT IS A TYPE OF IOT DEVICE THAT CAN RESPOND TO VOICE COMMANDS, COMPLETE BASIC TASKS, AND MIMIC THE STRUCTURE OF HUMAN CONVERSATION.

Adding variability to responses

When responses are hard coded, Alexa responds to user prompts with the same answer every time. To add variability, we used the Google Sheets and Google Drive APIs to make our spreadsheet an answer bank that Alexa could read from. We listed each possible response in individual cells and assigned them group numbers based on what question they correlated to. We iterated through cells based on group number and picked a random response for Alexa to output. This allowed Alexa to give different responses for the same user prompt.

As we added more responses to our spreadsheet, the skill would often timeout or be unable to loop through too many lines of the sheet. When looping through lines of the spreadsheet in an intent handler, a timeout occurs after 65–70 lines. When the loop includes actions, such as copying or writing, a timeout occurs after around 30 lines. Additionally, when reading the cells of two columns within the loop, the limit is reached at 15–20 lines. To optimize the performance, we put all responses in a single cell, delimited by hash marks and numbers. We read this cell at the beginning of each skill execution and parsed the responses within the individual intent handlers. This allowed us to bypass looping through the rows of the spreadsheet and the timeout issues.

We created a Google Apps Script in Google Sheets to automatically format the plain text content directly inputted from health-care educators into the optimized format needed for the skill. This permitted the user to keep the customizability of his or her content while enabling him or her to remain separated from the inner workings of the skill.

Finally, to keep track of Alexa's current responses, we created a second sheet within our spreadsheet and wrote to the sheet every time Alexa outputted a response. This let us easily see what Alexa said last and keep track of the delivered responses.

Timer-triggered responses

We investigated having Alexa produce a response that was not prompted by a user utterance. We chose to have Alexa "sneeze" when the user is 2 min into skill execution by taking note of the starting time of the skill and comparing it to the current time in each intent handler. When the starting time and the current time were 2 min apart, we had Alexa output a prerecorded sneeze audio from the Alexa Skills Kit Sound Library.

Keeping score

We explored how to keep track of which questions the user asked by creating a session and persistent attributes within our skill, which allowed us to store data between intents and skill sessions. Each intent had associated attributes that incremented each time a user asked that specific question. Furthermore, we created a global score attribute that was an aggregate of all the other scores. This let us keep track of which questions the user had or had not asked.

We added weighted scores for individual intents to enable the user to make some questions more important than others. We allowed some intents to increment their scores by more than one each time they were triggered, resulting in some intents contributing more to the aggregate score than others, thereby creating weighted scores.

Additionally, we created a function that permits the user to save his or her score for a single run of the skill. We created a third sheet within our spreadsheet that acted as a database of user scores. When a user asks to save his or her score, the values of each attribute are written to the sheet. These user scores are associated with the name of the user, which the user is prompted to say at the beginning of skill execution. Each row in the database corresponds to an individual run of the skill, meaning that all entries are unique regardless of duplicate usernames. This provides for the concept of user accounts, where the user can have a name and score associated with his or her attempt, which can be saved for later access.

Finally, providing custom performance feedback and after-action review (AAR) to users is essential for training. We structured the feedback to revolve around individual and aggregate scores. If the score for a question was greater than zero, the user had successfully asked that question and received positive feedback. Likewise, if the score for a question was zero, the user failed to ask that question and received negative feedback. The AAR was stored in the spreadsheet in the same one-cell formatting as the responses. Educators can input the general feedback topic in the spreadsheet, such as "medical history," and the skill will tell the user whether they successfully covered that topic based on the score for that intent and his or her overall score.

Master spreadsheet

Educators need to be able to input their own custom content and save individual user scores. We created a master spreadsheet that contains four-digit codes and links to another spreadsheet to handle multiple people or organizations using the same skill. Each code-link pair corresponds to a different organization, allowing each party to have his or her own spreadsheet to input content in and use for the skill. When the skill begins execution, the user is prompted to say his or her four-digit code. The skill then searches the master spreadsheet for the corresponding spreadsheet link and uses that new spreadsheet for skill execution. The master spreadsheet provides educators with their own separate spreadsheets, which have customizable content and an exclusive database for user scores.

DISCUSSION

The outcome of our exploration into the capabilities of Alexa and the Google APIs is a flexible and customizable review of systems' skill that allows users to ask personalized questions to an Amazon Alexa. The Google Sheets API connection enabled us to use a spreadsheet within the skill as an answer bank to add variability to Alexa's responses as a database to store user scores during skill execution and to keep track of Alexa's current response. Within the spreadsheet, we optimized reading and writing from the sheet by inserting all responses into one single cell to circumvent timeout problems. The skill keeps track of which questions the user has and has not asked as well as provides the user with the opportunity to save his or her score to a database and receive customized performance feedback. We also enabled a timer-triggered Alexa response, which caused Alexa to "sneeze" during skill execution. Finally, educators can have their own individualized spreadsheets and databases, which can be navigated to through the master spreadsheet.

IPAS COME WITH A WIDE RANGE OF BUILT-IN SKILLS, BUT THEY ARE OFTEN INFLEXIBLE AND DIFFICULT TO CUSTOMIZE.

Limitations

Limitations between Alexa and the Google APIs include Google Sheets imposing a limit of 50,000 characters per cell and request limitations (namely, 500 requests per 100 s per project, and 100 requests per 100 s per use). In addition, there are limitations when iterating through lines of the spreadsheet that depend on what is happening in the body of the loop. Actions such as copying and writing are more costly than simply reading through lines. Finally, the master spreadsheet has a limit of approximately 35–37 times, meaning that the skill can read and open a new spreadsheet link around 35–37 times before it times out. When a user code is implemented or the loop is storing the values of each cell, the limit tightens to 20–25 lines.

Future work

The implications of this technology are very widespread. With a lack of IPAs in the health-care field, technology that allows users to customize content is extremely valuable. Future plans include testing this technology with a user study to explore usability with health-care educators, and training effectiveness for health-care students. Additionally, we are interested in trying to combine this Alexa skill with the Unity platform to create a virtual health-care simulation that involves both graphics and speech.

We explored the technical capabilities of Amazon's Alexa in combination with Google Sheets and Google Drive APIs to create a customizable health-care training model. We created a custom skill that allows the user to input personalized dynamic content, initiate questions, add variability to Alexa's responses, keep a database of user scores, have time-triggered responses within skill execution, and receive feedback on his or her performance. The skill was designed as a review of systems' training models for health-care students; however, it has many additional possible applications in the health-care field. This technology explores new functions of intelligent personal agents in the health-care training industry.

PROVIDING CUSTOM PERFORMANCE FEEDBACK AND AFTER-ACTION REVIEW TO USERS IS ESSENTIAL FOR TRAINING.

REFERENCES

1. "*Voice assistants used by 46% of Americans, mostly on smartphones*," Pew Research Center, Aug. 25, 2020. [Online]. Available: https://www.pewresearch.org/fact-tank/2017/12/12/nearly-half-of-americans-use-digital-voice-assistants-mostly-on-their-smartphones/ (accessed Sep. 9, 2021).
2. T. Ammari, J. Kaye, J. Y. Tsai, and F. Bentley, "Music, search, and IoT: How people (really) use voice assistants," *ACM Trans. Comput.-Human Interact.*, vol. 26, no. 3, pp. 1–28, Apr. 2019, doi: 10.1145/3311956.
3. L. Laranjo et al., "Conversational agents in healthcare: A systematic review," *J. Amer. Med. Inform. Assoc.*, vol. 25, no. 9, pp. 1248–1258, Jul. 2018, doi: 10.1093/jamia/ocy072.
4. G. Riccardi, "Towards healthcare personal agents," in *Proc. Workshop on Roadmapping the Future Multimodal Interact. Res. Including Bus. Opportunities Challenges (RFMIR '14)*, 2014, pp. 53–56, doi: 10.1145/2666253.2666266.
5. B. Kinsella, "*Voice assistant consumer adoption report for healthcare 2019*," Voicebot.ai, Mar. 3, 2020. [Online]. Available: https://voicebot.ai/voice-assistant-consumer-adoption-repor-for-healthcare-2019/ (accessed Sep. 9, 2021).
6. M. Campbell and M. Jovanovic, "Conversational artificial intelligence: Changing tomorrow's health care today," *Computer*, vol. 54, no. 8, pp. 89–93, Aug. 2021, doi: 10.1109/MC.2021.3083155.
7. "Create Alexa skills kit | Amazon Alexa voice development," Amazon. [Online]. Available: https://developer.amazon.com/en-US/alexa/alexa-skills-kit (accessed Sep. 9, 2021).
8. *Google Cloud Platform*. [Online]. Available: https://console.developers.google.com/ (accessed Sep. 9, 2021).
9. A. Burnashev, *gspread*. [Online]. Available: https://docs.gspread.org/en/latest/ (accessed Sep. 9, 2021).
10. J. Hallman, "*Amazon Alexa skill offers supportive care to breast cancer patients*," Pennsylvania State Univ., University Park, PA, USA, Apr. 27, 2021. [Online]. Available: https://tinyurl.com/2wk79dd7
11. L. Dyrda, "ChristianaCare launches Alexa skill for home health: 4 Details," *Beckers Hospital Rev.*, Oct. 21, 2020. [Online]. Available: https://tinyurl.com/4ruzxc6r

SARAH JANE MEE is a computer science and information technology student at New Jersey Institute of Technology, Newark, New Jersey, 07103, USA. Contact her at sm2778@njit.edu.

SALAM DAHER is an assistant professor of informatics and computer science at New Jersey Institute of Technology, Newark, New Jersey, 07103, USA. She is also a courtesy faculty member in the College of Nursing at the University of Central Florida, Orlando, Florida, 32816, USA. She is the corresponding author for this article. Contact her at salam.daher@njit.edu.

EDITORS: Munindar P. Singh, singh@ncsu.edu
Pradeep Murukannaiah, p.k.murukannaiah@tudelft.nl

This article originally appeared in Internet Computing vol. 25, no. 6, 2021

DEPARTMENT: INTERNET ETHICS

Ethical Online AI Systems Through Conscientious Design

Pablo Noriega, IIIA-CSIC, 08193 Bellaterra, Spain
Harko Verhagen, Stockholm University, 114 19 Stockholm, Sweden
Julian Padget, University of Bath, Bath, BA2 7AY, U.K.
Mark d'Inverno, Goldsmiths, University of London, London, SE14 6NW, U.K.

There is increasing interplay between humans and artificial intelligent (AI) entities in online environments. With the growing autonomy and sophistication of these AI systems, the hybrid communities which are formed start to behave like the more familiar, human-only social systems. This sets up the challenge to find systematic ways to ensure reliable governance for these interactions just as we do in human communities. This article proposes a novel approach to build governance for hybrid communities using what we call Conscientious Design (CD). There are two key aspects to CD: 1) the introduction of value categories that guides the identification of relevant stakeholder values, coupled with 2) a tripartite model for online institutions that serves to describe the interactions of hybrid communities of humans and artificial entities in a way that is consistent with the values of all stakeholders.

We are entering a time of increasing interaction with artificial intelligent systems (AIS) in online environments. Moreover, these interactions will be increasingly complex as we become more familiar with inhabiting such communities and the artificial systems themselves become more sophisticated and autonomous. This suggests a pressing need to explore approaches to the design of such systems to build confidence that the emerging online environments and behaviors are places we would wish to inhabit.

The greater autonomy of artificial entities means increased potential for them to influence the social and psychological states of human participants. This growing potential raises new concerns about how one can protect participants' well-being when these more complex computational agents can be more incompetent, untrustworthy and—even—malevolent.

1089-7801 © 2021 IEEE
Digital Object Identifier 10.1109/MIC.2021.3098324
Date of current version 6 December 2021.

While engineering ethical considerations into AIS are often spoken about, current practice is patchy at best. Even if ethics are considered, there are no systematic or principled means to ensure that the design and implementation of a system with convincing answers to questions such as: what does it mean to do the "right" thing?, how can it be known with any degree of certainty that a new AIS will support the "right" thing?, and when is enough "enough" in terms of what needs to be thought about?

Furthermore, the risks of getting it "wrong," and a new system causing harm, are hard to assess too. Not only because all kinds of unplanned behaviors and impacts could emerge, but also because of a lack of documented experience in addressing ethical concerns in AIS design. Because considering these factors together is so hard, it might lead to ignoring the issue altogether or hoping that basic common sense will be enough to resolve any problems on the fly.

In response to these concerns, we have developed the notion of *Conscientious Design (CD)* as a systematic and practical approach to support practitioners in the ethical design of AIS. It is an approach that builds on well-established practices in value-sensitive design

(VSD),[1] Alexander's "habitable spaces",[2] and Deming's total quality management (TQM).[3] It also provides a way of using familiar agile concepts to imbue values in AISs. Additionally, it puts human and artificial agent participants in control of coevolution of the online spaces they jointly inhabit.

Participants in Alexander's habitable spaces are physically constrained whereas in online systems they are constrained in different ways. First, they are constrained by the platform itself and what actions it allows, known as platform-provided affordances[4] (e.g., "buy," "like," "ban"). Actions not provided by the platform simply cannot take place. Second, actions of one participant are constrained by the normative expectations that the other participants have of what is acceptable or unacceptable behavior (e.g., spamming, helping, ignoring), where noncompliance may lead to sanctions against the acting agent. These two categories of constraint are perhaps most easily understood through our own experiences of using online platforms (e.g., shopping, social networks).

We base our proposal for CD on a particular subclass of AIS that we have been researching for some years, called *online institutions (OIs)*.[5,6] OIs contain policies that facilitate the governance of participant activity, either through what a participant is allowed to do in certain circumstances or what a participant may choose to do or not to do for the sake of any social consequences. Online institutions embody both affordances and norms, interpreting Alexander's "timeless way of building" for the social—often commercial—spaces in which we participate on the Internet. Furthermore, OIs (as with all AISs) are software constructs, and so have an intrinsic adaptability and resilience, which means that they can in theory support Deming's evolutionary approach to the achievement of quality over time, founded on VSD's value principles. We also take the position that by considering online institutions we can most effectively map out the principles and building blocks of conscientious design, which can then be applied to a wider class of AIS in due course.

CONSCIENTIOUS DESIGN

Stakeholders in VSD are presented with a simple ethical framework: first consider what is right, and secondly what is good,[1] which hints at a hierarchy of values and debates over which values are right and which are good. This creates two challenges: how to identify the (small) set of core values and to which value or values to associate different aspects of the design, without connecting everything to everything.

CD builds on VSD by providing a value framework from within which to argue about the "how and why" of stakeholder values, rather than whether one value is more important than another. The framework involves three *value categories* (thoroughness, mindfulness, and responsibility), a *systematic identification of contexts* (through the *WIT pattern*) where these categories are instantiated in OIs, and a process to *make values operational*.

The CD value categories are as follows:

- **Thoroughness**: This refers to conventional technological values that promote the technical quality of the system. In any (standalone) system, values include completeness and correctness of the specification and implementation, reliability and efficiency of the run-time version of the system, robustness, resilience, accessibility, and security. Thoroughness also applies (in the "situated OI") to the technological compatibility of the OI with the context where it is embedded, as well as its integrity (intrusions and data or communication corruption);
- **Mindfulness**: We have chosen this word carefully to respond to the considerations about impact on human users that are so often overlooked. In its characterization, mindfulness includes building a wider awareness of what is happening to humans and society through the use of technology to guide us in making the right choices, in line with Deming's principles. Examples of values in this category concern data ownership (privacy, data agency, usage traces), and well-being (accessibility, respect of users' attention).
- **Responsibility**: These are values that address the effects toward the owner, the users, and external stakeholders (regulators, suppliers, partners) of using the OI. Here, we can also include the effects of the system on the context in which it is situated (liability, accountability), and how that context may affect intended users, designers, and owners (legitimacy, user protection, no hidden agency).

In broad terms, CD's contributions are the distinctive attention to policies that govern interactions the systematic separation of analysis by stakeholder, context, and time supported with the WIT pattern. In particular, the CD proposal supports the initiatives from the EU[7] and IEEE[8] on building AIS. Indeed, these initiatives underline the timeliness of CD. In Table 1, we illustrate how CD values relate to the EU and IEEE principles, respectively, based on the keywords used in the documents in which they are described. For

TABLE 1. Mapping EU[7] and IEEE[8] principles onto the three CD value categories thoroughness, mindfulness, and responsibility. Italics denote examples of operationalized values and plain text the indicators of these values.

		Thoroughness	Mindfulness	Responsibility
EU HLEG Guidelines for Trustworthy AI Ethical Principles	**Human autonomy**	*respect of user's preferences* flexible service options unobtrusive interface	*respect of user's attention* quick trip negotiation	*respect of user's freedom* fair contract termination clauses
	Prevention of harm	*risk minimisation* risk / liability driver's qualifications	*certainty of commitments* assurance of identities of users before trip	*compatibility with local culture* reliable and convenient payment options
	Fairness	*user neutrality* uniform role-based criteria	*fair service allocation* best car best time	*fair fares* fair and competitive fares
	Explicability	*thorough representation* enough indicators for ongoing value assessment	*reliable and accessible use relevant information* pricing algorithm should explain itself	*legal and fiscal responsibility* clear invoices
General Principles of Ethically Aligned Design for Autonomous / Intelligent Systems	**Human rights**	*bias free* good quality system-generated data	*protection of users' rights* drivers get paid	*environmentally responsible* vehicle emissions requirements
	Well-being	*integrity of user identity* user access and service completion validation	*user satisfaction* driver compensations and rewards	*passenger and driver safety* insurance
	Data agency	*integrity of system data* removal of app implies removal of server-side data	*clear and enforceable data use policies* user's choice over user-associated data ownership	*fair third-party data access* foreclosure diagnostics
	Effectiveness	*solid payment methods* credit verification/application	*quick negotiation* certainty about mutual commitments	*liability protection* insurance
	Transparency	*adequate value assessment indicators* elicit WIT-contexts stakeholders' values	provide relevant and accurate information keep user- relevant history	*compatibility with social norms* data ownership policies
	Accountability	*reliable value attainment assessment* indicators for historic and ongoing assessment	*availability of user-relevant and accessible information* passenger's / driver's activity history	*compliance with local regulations* auditable transactions
	Awareness of Misuse	*reliable interfaces* thorough input/output checkpoints	*protection of users needs* user-satisfaction elicitation	*awareness of potential illegal activity* collusion among car-owners
	Competence	*objective performance assessment* correct app updating	*alignment with business objectives* cost/benefit analysis cash flow	*technological compatibility* accurate mapping of available cars

instance, the EU Guidelines have under the ethical principle of explicability the following example measures: "traceability, auditability, and transparent communication on system capabilities".[7] These belong to the CD value of responsibility, in that they describe the anchoring of the system. As an example of mapping IEEE ethical design principles, consider competence. This addresses safe and effective operation,[8] i.e., it belongs to the CD value of thoroughness, with its focus on the technical quality of the system.

Apart from showing how to map all EU[7] and IEEE[8] principles onto the CD proposal, Table 1 also shows that these principles can be mapped onto all CD value categories. Thus, CD value categories support more than one way of looking at each particular principle. This is a notable benefit of CD's principled approach.

ONLINE INSTITUTIONS AND THE WIT PATTERN

CD aims to help designers in debating why and what of the system, but the translation from "what" to "how" needs equally careful handling to maintain the separations of concerns suggested above. As with Alexander's blueprints, the objective is not to provide an answer, but a way to think about the answer and arrive at an appropriate solution every time. Therefore, we propose the World-Institution-Technology design pattern for OIs (see Figure 1), where the world (W) is a social space that is a subcontext of the real world, institutions (I) are the policy frameworks into which the values that characterize the system are imbued, and the technological space (T) where online interactions are processed according to software representations of the institutional conventions.

Online institutions in CD are the glue that binds W, I, and T together, to mirror the functions of conventional social and economic institutions.[9] This subclass of sociotechnical systems is formally defined in Noriega *et al.*[10,11] and is a refinement of other abstractions of systems for social coordination and artificial or electronic institutions.[5,12] Informally, an OI provides technological support for human and software agents to interact online with each other, and establishes the policy—the "rules of the game"—that governs those interactions. The terms of the policy determine what fragment of the real world is relevant, what events and actions that take place in the world are recognized by the institution and what their effects in the institution are, and vice versa. For this purpose, an OI 1) maintains an institutional state that is accessible to all the active participants and 2) may recognize whether an action is correct (in the prevailing circumstances) and, if so, update the institutional state accordingly. For example, if a customer signals—via the *Uber* app—that a pick-up is not taking place, the system would ignore

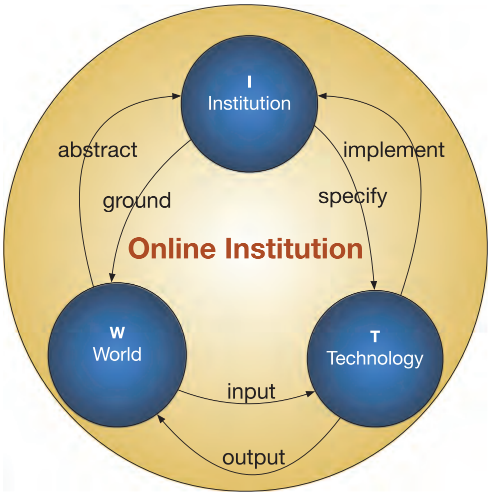


FIGURE 1. The WIT Tripartite Pattern: the World, Institution, and Technology Views and the relationships between them (after[13]).

the signal if a driver is about to arrive or notify the customer that another driver is on their way.

We now look in more detail at the relationships between the components of the WIT pattern (Figure 1):

- W↔I: intuitively, I is an abstraction of the relevant subcontext in W, which captures "just enough" of the real-world dynamics—the actions and events that can occur that matter for the subcontext, like movement, or picking up or dropping items in a game—and an institutional model that represents the policy that applies to those recognized actions and events. In the other direction, institutional changes need grounding to have consequences in the social (world) context, such as a passenger rating affecting driver selection in *Uber*.
- I↔T: the abstraction in I provides the specification for what must happen in T, telling the developer what function the technology space should deliver, while the relationship in the other direction documents how the technology space implements what I specifies.
- W↔T: the relationship between W and T that enables the participants of the social (world) context to interact, by whatever interfaces are appropriate (webpages, phones, game handsets, VR, sensors of various kinds) providing inputs to the OIs (actions and events) and receiving outputs (institutional interpretations and consequences of those actions and events).

Moreover, the WIT pattern also helps to differentiate what to consider when examining the legal, social, and technological compatibility of the OI as a system that is situated in its (evolving) working environment.[11]

PUTTING VALUES INTO PLAY

Values are powerful and practical devices to imbue ethical behavior in AIS. In general, values help to assess the "worthiness" of a state of affairs and to decide the "right" action to take.[14] Institutional governance should promote or require actions whose effects align with stakeholders' values and prevent or discourage those that do not. We propose a three-stage process for making values operational:

1. Value interpretation consists of identifying behaviors and outcomes that are characteristic of that value so that these are encouraged or guaranteed to happen. The three CD value categories must be instantiated with concrete values that allow for a refinement of its interpretation, implementation, and assessment. Interpretations of the same value may vary depending on the context in which the behavior is to be observed, the perspective of the stakeholder who observes it, and the moment when the value is assessed.

The WIT pattern facilitates this analysis with the identification of the different contexts. There are two approaches for defining the meaning of a value. One is to produce an explicit description of behaviors that uphold (or demote) the value, the other is to choose a set of indicators—observable parameters in the state of the system—that reflect support for the value (or its demotion).

2. Value implementation can be achieved by focusing on the behaviors and outcomes aligned with the value. There are three typical strategies of implementing values. They are not mutually exclusive and strategy selection is a design decision. The three strategies are as follows:

- *Hard-wire constraints* and procedures that implement specific behavior and indicators associated with the interpretation of values. This presupposes the choice of the relevant entities that provide the basis for the institutional model and its implementation. This hard-wiring needs to adapt to the evolution of an OI. For instance, in online multiplayer games such as League of Legends, the base capacities and skills of the characters the players can choose from are given, as are the ways in which these can be extended during game-play.
- *Use explicit policies* (e.g., technical norms).[15] These may comprise 1) functional norms that

specify the preconditions and the effects of admissible actions; and can be easily linked with indicators; or 2) procedural norms that define how to perform and implement a specific behavior that interprets a value, for example, in *Uber*, a "fairness" norm assigns a rider the closest available car but prioritizes cars with higher client satisfaction ratings.

- *Influence the decision-models of participants* by providing additional information or arguments that may promote a change of decisions. In online games, such as League of Legends, the problem of toxic gaming and inappropriate language between temporary teammates is detrimental to enjoyment. In League of Legends, at first a sanctioning strategy was chosen—initially using selected human players as a jury to judge complaints,[16] later replaced by an automated sanctioning system which was criticized, among other reasons, for not being transparent.[17] In its latest incarnation, a positive reward system has been put in place as an honor system in which team mates can give each other positive feedback. How this feedback is represented in the game (a badge with a numerical value) and what it may result in (extra in-game rewards) has changed over time but an overall critique remains to this system as well: it is the game company who decides what is and what is not transgressing the "honor rules of the game",[17] i.e., not all stakeholders are part of the discussion on how to assess the fulfilment of the value of "fun".

3. Value assessment determines to what degree a value is being attained. This may either be validating that a required behavior is achieved, or measuring value indicators. Since value interpretation (and implementation) is "context dependent" we put all the needed assessment components into a *Value Assessment Framework* that, for each stakeholder, consists of: 1) the values that are relevant in the specific assessment context; 2) the corresponding interpretation and validation/measuring mechanisms for each value; and 3) the aggregation function for the set of values.

CONCLUDING REMARKS

We are all aware of increasing interaction in online communities of human and software participants. Many of these have been designed and implemented without truly recognizing that a new kind of responsibility in software design is needed to protect human well-being.

In this article, we have outlined conscientious design (CD) as our response to this need. We have specifically applied CD to *online institutions* which are a subclass of AISs, where governance is explicitly represented and enforceable.

Our intention in proposing CD is to support developers of ethical hybrid online social systems in three ways:

- First, to provide a blueprint for the construction of online systems that we would be happy to inhabit. This blueprint is achieved through the separation of world, institution, and technological concerns using the WIT pattern to facilitate the design of online institutions.
- Second, we propose three value categories—thoroughness, mindfulness, and responsibility—and provide a characterization and justification for each. These provide a high-level guide to embedding the shared and agreed values of stakeholders in OIs. We believe that any consideration of ethical issues should consider these three categories in detail.
- Third, to enable the design of explicit, transparent governance mechanisms that contain mutually comprehensible representations of human authored policies to say what participants may do under what circumstances.

We believe these considerations together support the explicit consideration of ethical aspects. They enable stakeholders, including designers, to explicitly introduce their own values into the design of ethical AISs. It enables a balanced focus on the affordances and norms that are so critical in understanding governance. CD enables the system to adapt transparently as the needs and value priorities of stakeholders change over time.

In closing we set out why the CD approach matters:

- CD is *principled:* It provides an intuitive way to operationalize the principles set out in the trustworthy AI[7] and ethically aligned design[8] guidelines.
- CD *reorients existing methods* for AIS. CD extracts elements from VSD, design patterns, and process quality to apply known thinking from agile development to target a class of internet-based systems.
- CD is *timely* because we are in the early stages of the construction of online sociotechnical systems that have both human and AIS participants.

- CD is *practical:* value imbuing is not a trivial process but our experience shows that it can be tackled with a principled strategy that interprets conscientious values in relevant contexts (stakeholders, stand-alone, situated) and uses adequate devices for making them operational (value interpretation, instrumentation, measurement, aggregation).
- CD is *malleable*. It requires an ongoing implementation process involving stakeholders from the start. Values are not set in stone; with CD, they are identified and fit (ex-ante) to the specific context and are progressively assessed and adapted ex-post.
- CD facilitates *continuous improvement*—as modifications or add-ons—for refactoring conscientious values into existing systems.

We hope this work can be the start of building an interdisciplinary community of researchers and practitioners who can join forces to further develop the body of CD practice with rigorous descriptions of (reusable) CD components, documenting use cases that embed ethical considerations in the design process, and so build better, fairer, and safer online worlds.

REFERENCES

1. B. Friedman, D. G. Hendry, and A. Borning, "A survey of value sensitive design methods," *Found. Trends Hum.-Comput. Interact.*, vol. 11, no. 2, pp. 63–125, 2017.
2. C. Alexander, *The Timeless Way of Building*, vol. 1. New York, NY, USA: Oxford Univ. Press, 1979.
3. W. Edwards Deming, *Quality, Productivity, and Competitive Position*. Cambridge, MA, USA: MIT Press, 1982.
4. J. J. Gibson, "The theory of affordances. The ecological approach to visual perception," Houghton Mifflin (Boston), pp. 127–143, 1979.
5. M. d'Inverno, M. Luck, P. Noriega, J. A. Rodriguez-Aguilar, and C. Sierra, "Communicating open systems," *Artif. Intell.*, vol. 186, pp. 38–94, 2012.
6. P. Noriega, H. Verhagen, M. d'Inverno, and J. Padget, "A manifesto for conscientious design of hybrid online social systems," in *Proc. Int. Workshop Coordination, Org., Inst., Norms Agent Syst. XII Revised Sel. Papers*, 2016, Springer LNCS, vol. 10315, pp. 60–78.
7. High-Level Expert Group on AI (AI HLEG), "Ethics guidelines for trustworthy AI," Brussels: European Commission, Apr. 2019, doi: 10.2759/346720.
8. The IEEE Global Initiative on Ethics of Autonomous and Intelligent System, "Ethically aligned design: A vision for prioritizing human well-being with autonomous and intelligent systems," First ed. 2019. Accessed: Oct. 4, 2019. [Online]. Available: http://standards.ieee.org/content/ieee-standards/en/industry-connections/ec/autonomous-systems.html
9. D. North, *Institutions, Institutional Change and Economic Performance*. Cambridge, U.K.: Cambridge Univ. Press, 1991.
10. P. Noriega, J. Padget, H. Verhagen, and M. d'Inverno, "Towards a framework for socio-cognitive technical systems," in *Coordination, Organizations, Institutions, and Norms in Agent Systems X*, Springer LNCS 9372, 2014, pp. 164–181.
11. P. Noriega, J. Padget, and H. Verhagen, "Anchoring Online Institutions," in *Anchoring Institutions. Democracy and Regulations in a Global and Semi-Automated World* (P. Casanovas and J. J. Moreso, Eds.), New York, NY, USA: Springer, 2021. In press.
12. H. Aldewereld, O. Boissier, V. Dignum, P. Noriega, and J. Padget, eds., *Social Coordination Frameworks for Social Technical Systems*, vol. 30 in Law, *Governance Technol. Ser.*, New York, NY, USA: Springer, 2016, doi: 10.1007/978-3-319-33570-4.
13. R. Christiaanse, A. K. Ghose, P. Noriega, and M. P. Singh, "Characterizing artificial socio-cognitive technical systems," in *Proc. Eur. Conf. Social Intell.*, 2014, pp. 336–346.
14. S. H. Schwartz, "Universals in the content and structure of values: Theoretical advances and empirical tests in 20 countries," *Adv. Exp. Social Psychol.*, vol. 25, pp. 1–65, 1992.
15. I. van de Poel, "Embedding values in artificial intelligence (AI) systems," *Minds Mach.*, vol. 30, no. 3, pp. 385–409, 2020.
16. M. Johansson, H. Verhagen, and Y. Kou, "I am being watched by the tribunal: Trust and control in multiplayer online battle arena games," in *Proc. Found. Digital Games*, 2015. [Online]. Available: http://www.fdg2015.org/papers/fdg2015_paper_37.pdf. Accessed 2021-08-03
17. S. Tomkinson and B. van den Ende, "'Thank you for your compliance': Overwatch as a disciplinary system," *Games Culture*, 2021, doi: 10.1177/15554120211026257.

PABLO NORIEGA is a Tenured Scientist in the Artificial Intelligence Research Institute, Spanish National Research Council (IIIA-CSIC). His research interests are mostly in regulated multiagent systems and agreement technologies. Lately, he has been active in modeling value-driven policy making. Contact him at pablo@iiia.csic.es.

HARKO VERHAGEN is a Senior Lecturer in Computer and Systems Sciences at Stockholm University. He has been working on combining social science theories and models with agent research, and applying these in social simulation as well computer game studies since the early 1990s. Contact him at verhagen@dsv.su.se.

JULIAN PADGET is a Reader (Associate Professor) in Artificial Intelligence with the University of Bath, Bath, U.K. He has been working on norm representation and reasoning for multiagent systems since the mid 1990s, while exploring applications in virtual reality, generative narrative, policy-making and legal reasoning. He is the corresponding author of this article. Contact him at j.a.padget@bath.ac.uk.

MARK D'INVERNO is a Professor of Computer Science at Goldsmiths, University of London, London, U.K. He was formerly Pro-Warden Research and Enterprise and subsequently Pro-Warden International. He has been PI and Co-I in a range of U.K. and internationally funded projects and has worked on the interface of AI, social science, and arts practice for several decades. Contact him at dinverno@gold.ac.uk.

EDITOR: Hsiao-Ying Lin, IEEE Member, hsiaoying.lin@gmail.com

DEPARTMENT: ARTIFICIAL INTELLIGENCE/ MACHINE LEARNING

This article originally appeared in Computer vol. 55, no. 1, 2022

Assessing AI Fairness in Finance

Lachlan McCalman and Daniel Steinberg, *Gradient Institute*
Grace Abuhamad and Marc-Etienne Brunet, *ServiceNow*
Robert C. Williamson, *University of Tübingen*
Richard Zemel, *University of Toronto*

If society demands that a bank's use of artificial intelligence systems is "fair," what is the bank to actually do? This article outlines a pragmatic and defensible answer.

Artificial intelligence (AI) systems are becoming ubiquitous in a diverse and ever-growing set of decision-making applications, including in the financial sector. AI systems can make consequential decisions at a speed and volume not possible for humans, creating new opportunities to improve and personalize customer service but also increasing the scale of potential harm they can cause if they are misdesigned.

AI systems unfairly discriminating against individuals by their race, gender, or other attributes is a particularly common and disheartening example of this harm. For example, soon after launching its credit card partnership with Goldman Sachs in 2019, Apple had to investigate its system for gender bias. This bias, if left unchecked, could have limited women's access to credit, harming those potential customers and increasing risks of regulatory noncompliance for the business.[1] However, there is no simple solution to preventing these kinds of incidents: helping AI live up to its promise of better and fairer decision making is a tremendous technical and social challenge.

One of the key design mistakes behind harmful AI systems in use today is an absence of explicit and precise ethical objectives or constraints. Unlike humans, AI systems cannot apply even a basic level of moral awareness to their decision making by default. Only by encoding mathematically precise statements of our ethical standards into our designs can we expect AI systems to meet those standards.

Technical work to develop such ethical encodings is burgeoning, with much of the focus on the fairness of AI systems in particular. This work typically involves developing mathematically precise measures of fairness suitable for designing into AI systems. Fairness measures use the system's data, predictions, and decisions to characterize its fairness according to a specific definition (for example, by comparing the error rates of the system's predictions between men and women). The exercise of defining fairness in mathematical terms has not "solved" fairness but rather surfaced the complexity of the problem at the definitional stage. There now exists a panoply of fairness measures, each corresponding to a different notion of fairness and potentially applicable in different contexts.

Parallel to the work of encoding ethical objectives mathematically is a broader social effort to develop principles and guidelines for ethical AI. These aim to help the designers, maintainers, and overseers of AI systems recognize and ameliorate ethical risks. Governments, corporations, and other organizations have released hundreds of such frameworks in the last few years, many with common themes like the importance of explanations of an AI system's decisions, the need to provide mechanisms for redress when errors occur, and the need to understand and minimize avoidable harms caused by the system. For example, the National Institute of Standards and Technology released a proposal last year to identify and manage bias using a three-stage approach.[2]

Digital Object Identifier 10.1109/MC.2021.3123796
Date of current version: 12 January 2022

However, a gap remains between the technical efforts and the broader design principles. Designers building AI systems have access to principles, on the one hand, and mathematical tools, on the other, but little guidance about how to integrate these two resources and build a system that utilizes them in consequential settings.

THE FEAT PRINCIPLES AS A STARTING POINT

Financial services institutions (FSIs) manage billions of dollars' worth of transactions per day and are increasingly adopting AI solutions as part of this business, including for determining loan and credit card approvals, conducting marketing, and detecting fraudulent behavior. In particular, the Massachusetts Institute of Technology *Technology Review Insights* found that businesses in the Asia-Pacific region are quicker to adopt AI systems than any other part of the world.[3] The scale and importance of these systems to FSI's daily operations means that if they are misdesigned, they can create reputational, operational, and legal risks for businesses and unnecessary harms to customers.

To begin addressing the ethical risks of AI decision making in finance and in doing so encourage AI adoption, the Monetary Authority of Singapore (MAS) released principles for responsible AI in the finance industry.[4] These "FEAT Principles" (Fairness, Ethics, Accountability, Transparency) were developed in partnership with Singaporean and international financial institutions and AI experts, known as the Veritas Consortium,[5] and describe aspirational ethical properties that an AI system would have, such as not systematically disadvantaging individuals, or groups, without justification.

While appearing simple, these principles contain within them complex and value-laden questions such as when a group or individual is being "systematically disadvantaged" and what data count as "relevant" for a particular application. Like the concept of fairness itself, these questions have no single uncontested answer, nor one that is independent of ethical judgment. Nor do the principles provide guidance for which (if any) of the myriad fairness measures developed may be appropriate to use to specify unjustified systematic disadvantage or unintentional bias.

FROM PRINCIPLES TO GUIDANCE: FEAT FAIRNESS ASSESSMENT METHODOLOGY

Since releasing the FEAT Principles, MAS and the Veritas Consortium have worked with teams of experts to develop implementation guidance. In January 2021, they released two white papers that detailed the first step in that implementation: a methodology for assessing AI systems for alignment with the FEAT Fairness principles[6] (with the other principles relating to Ethics, Accountability, and Transparency being tackled in a later phase) and a set of detailed case studies illustrating the application of the methodology to credit scoring and customer marketing systems.[7] We led the development of the methodology and the case studies as part of the core authorship team. The methodology comprises a set of questions (and accompanying guidance) answered by the people responsible for the AI system. Their answers go to an independent set of assessors that judge the system's alignment with the FEAT Principles.

The design of the methodology had to accommodate two critical but conflicting requirements: It had to be generic enough to be applicable across a whole industry and applicable to systems with different purposes, but specific enough to be useful and implementable by practitioners who may not be experts in algorithmic fairness or ethics. The final design of the methodology tries to balance these competing requirements with three key design pillars: asking users to stake their ethical claim, focusing on the harms and benefits of the system, and scaling the assessment to the system risk.

Asking System Owners to Stake a Claim

The first design pillar of the methodology is that it asks system owners to stake a claim on what they believe the fairness objectives of the system should be. Any assessment that can be applied to different AI systems cannot itself mandate specific notions or measures of fairness, such as the exact circumstances that constitute unjustified systematic disadvantage (see FEAT Principle 1). Different measures of fairness imply different ethical stances, and no methodology could hope to enumerate the right choice in every situation, nor impose a particular choice that aligns the designer's (or a particular community's) ethical stance.

In philosophical literature, fairness is known as an "essentially contested" concept. While the general notion of fairness is commonly understood, different people will have different ideas about exactly what is fair in a particular context. This also applies to the selection of precise fairness objectives that can be encoded into an AI system. For example, in a hiring scenario, both the application of gender quotas to remove the effects of past and current discrimination, as well as blind hiring in which the gender of applicants is obscured, are just two of many conflicting versions of fair hiring. Each of these approaches entails a different hiring process and will produce different results. Each has proponents and detractors, both with reasoned arguments that may depend on the details of the particular situation and the necessary choice of a baseline against which to compare. Deciding on a particular fairness measure for an AI system is akin to selecting one of these approaches to fair hiring; the use of a particular mathematical measure of fairness implies a specific set of ethical values and priorities.

Imposing particular fairness measures on a whole class of AI systems would certainly be ignoring the unique circumstances and context of each system as well as the ethical preferences of the people responsible for it. Therefore, the set of fairness measures can only be decided at a per-system level. Because no jurisdiction has yet developed regulation that mandates certain measures in certain circumstances (which may not even be possible or advisable), it must be the people responsible for that system that decide how its fairness should be measured.

The FEAT Fairness Assessment Methodology is built around this idea of the system owners "staking a claim" by stating their fairness objectives and how they're measured, preferably at design time. The assessment then asks them for evidence to convince an independent assessor that the system meets these objectives. This approach separates the question of "what is fair in this situation?" from the question of "does this system operate in accordance with its stated fairness objectives?" An expert can answer the second question with the output of the methodology. By sharing parts of the assessment with people affected by the system, independent ethics panels, external regulators, or the wider public, the answer to the first question can also be examined and critiqued.

Focusing on Harms and Benefits

The second design pillar of the methodology addresses the problem that to be useful, the methodology cannot simply offload all of the work of developing and measuring fairness objectives and constraints onto the users. To help in this task, the methodology asks system owners to analyze the harms and benefits that the system may create, and the different individuals and groups that it may impact. Once FSIs have identified these, they can develop fairness measures from them by estimating how these harms and benefits are distributed across the population. The resulting fairness measures may have already been developed in the literature or could be novel and specific to the system.

This approach inverts the common question of which fairness measure to choose? for an AI system: instead, it asks system owners to first decide who the system impacts and under what circumstances (noting that these choices also involve ethical judgment). Specific fairness measures can then be derived from the harms, benefits, and impacted people with guidance from the methodology.

However, understanding and developing measures for a system's impact is likely a substantial undertaking, especially when the impact may be indirect or difficult to measure. For consequential systems this effort is paramount, but for the potentially hundreds of small, proof-of-concept or research-style models used within an FSI, performing a full assessment may be an impossible workload.

Scaling for Risk

The third and final design pillar of the methodology addresses the workload involved in assessing the hundreds of AI systems in a large organization. It specifies that systems with greater risk, for example, that affect many people or that make consequential decisions, should be assessed in greater detail.

FSIs typically already undertake these kinds of risk-scaled model assessments but with a focus on financial harms. The methodology is designed to be incorporated into these processes, adding considerations of fairness risks for customers. The way it is integrated is not prescribed owing to how differently FSIs organize their internal processes, however, the methodology does make suggestions based on common model risk management approaches within FSIs.

NEXT STEPS TOWARD IMPLEMENTATION

To ensure that the final version of the assessment methodology was indeed useful and practical to implement, we applied it to a number of real and synthetic AI systems, releasing these as accompanying case studies.[7] The case studies focused on two application areas in which AI systems are commonly deployed: customer marketing and credit scoring.

Both use cases have fairness risks traditionally and deploying AI systems in these cases can amplify these risks or introduce new ones. Credit scoring has faced risks such as the consequential impact of decisions and managing evidence of historical discrimination. Marketing also risks harming vulnerable people when targeting products, such as promoting high-interest credit cards to compulsive spenders. For both credit scoring and marketing systems, the scalability and consistency of AI decision making exacerbate potential for systematic harm to groups of customers over others.

The Veritas Consortium has now reviewed assessment methodology, and members are likely to implement the assessment methodology internally. In 2021, work continued on assessments and guidance for the other FEAT Principles (the "Ethics, Accountability, and Transparency" parts) and case studies for AI systems used in insurance. These concepts are not independent of fairness, so we will likely see iteration of the fairness methodology and integration into a single, holistic assessment.

We hope that, while being voluntary, FEAT Fairness assessments will become common practice in the finance industry and that regulators around the world will study them carefully to stimulate and inform future guidelines and regulation. We also hope that institutions begin to publish some or all of their FEAT Fairness assessments, giving the wider community an ability to understand, and voice opinions on, these systems that make consequential yet currently opaque impacts on many people's lives.

ACKNOWLEDGMENTS

The FEAT Fairness Assessment Methodology was prepared and issued by the MAS, HSBC, UOB, EY, Element AI, Gradient Institute, and IAG Firemark Labs. Authors McCalman, Abuhamad, Steinberg, and Brunet served as part of the core team. Authors Williamson and Zemel served as academic advisors for the development of the methodology. This article reflects the authors' views, and the Monetary Authority of Singapore is not responsible for the information contained.

REFERENCES

1. T. Telford, "*Apple Card algorithm sparks gender bias allegations against Goldman Sachs*," The Washington Post, 2019. [Online]. Available: https://www.washingtonpost.com/business/2019/11/11/apple-card-algorithm-sparks-gender-bias-allegations-against-goldman-sachs/
2. R. Schwartz, L. Down, A. Jonas, and E. Tabassi, "*A proposal for identifying and managing bias in artificial intelligence draft*," NIST Special Publication 1270, 2021. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1270-draft.pdf
3. "*The global AI agenda: Asia-Pacific*," MIT Technology Review Insights, 2020. [Online]. Available: https://www.technologyreview.com/2020/04/23/1000335/the-global-ai-agenda-asia-pacific/
4. "*Principles to promote fairness, ethics, accountability and transparency (FEAT) in the use of artificial intelligence and data analytics in Singapore's financial sector*," Monetary Authority Of Singapore, 2018. [Online]. Available: https://www.mas.gov.sg/publications/monographs-or-information-paper/2018/FEAT
5. "*MAS partners financial industry to create framework for responsible use of AI*," 2019. https://www.mas.gov.sg/news/media-releases/2019/mas-partners-financial-industry-to-create-framework-for-responsible-use-of-ai

6. "*Veritas document 1 FEAT fairness principles assessment methodology*," Veritas Consortium, 2020. [Online]. Available: https://www.mas.gov.sg/-/media/MAS/News/Media-Releases/2021/Veritas-Document-1-FEAT-Fairness-Principles-Assessment-Methodology.pdf
7. "*Veritas document 2 FEAT fairness principles assessment case studies*," Veritas Consortium, 2020. [Online]. Available: https://www.mas.gov.sg/-/media/MAS/News/Media-Releases/2021/Veritas-Document-2-FEAT-Fairness-Principles-Assessment-Case-Studies.pdf

LACHLAN MCCALMAN is chief practitioner at the Gradient Institute, Canberra, 2006, Australia. Contact him at lachlan@gradientinstitute.org.

DANIEL STEINBERG is a principal researcher at the Gradient Institute, Canberra, 2006, Australia. Contact him at dan@gradientinstitute.org.

GRACE ABUHAMAD is an applied research scientist at ServiceNow, Montreal, H2S 3G9, Canada. Contact her at grace.abuhamad@servicenow.com.

MARC-ETIENNE BRUNET is an applied research scientist at ServiceNow, Montreal, M5G 1M1, Canada. Contact him at marc-etienne.brunet@servicenow.com.

ROBERT C. WILLIAMSON is the W3 Professor in Foundations of Machine Learning Systems in the Faculty of Mathematics and Natural Sciences, University of Tübingen, Tübingen, 72074, Germany. Contact him at bob.williamson@uni-tuebingen.de.

RICHARD ZEMEL is a professor of computer science at the University of Toronto and the cofounder and director of research at the Vector Institute, Toronto, M5G 1M1, Canada. Contact him at zemel@cs.toronto.edu.

IEEE Internet Computing
VOLUME 22, NUMBER 2
MARCH/APRIL 2018
IoT-Enhanced Human Experience

SUBMIT TODAY

IEEE TRANSACTIONS ON
SUSTAINABLE COMPUTING

SCOPE

The *IEEE Transactions on Sustainable Computing* (*T-SUSC*) is a peer-reviewed journal devoted to publishing high-quality papers that explore the different aspects of sustainable computing. The notion of sustainability is one of the core areas in computing today and can cover a wide range of problem domains and technologies ranging from software to hardware designs to application domains. Sustainability (e.g., energy efficiency, natural resources preservation, using multiple energy sources) is needed in computing devices and infrastructure and has grown to be a major limitation to usability and performance.

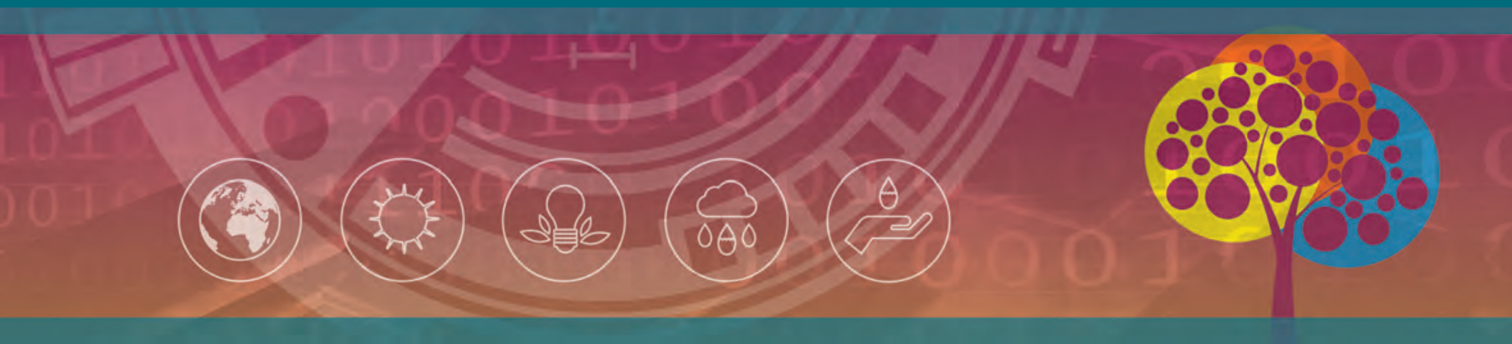
Contributions to *T-SUSC* must address sustainability problems in different computing and information processing environments and technologies, and at different levels of the computational process. These problems can be related to information processing, integration, utilization, aggregation, and generation. Solutions for these problems can call upon a wide range of algorithmic and computational frameworks, such as optimization, machine learning, dynamical systems, prediction and control, decision support systems, meta-heuristics, and game-theory to name a few.

T-SUSC covers pure research and applications within novel scope related to sustainable computing, such as computational devices, storage organization, data transfer, software and information processing, and efficient algorithmic information distribution/processing. Articles dealing with hardware/software implementations, new architectures, modeling and simulation, mathematical models and designs that target sustainable computing problems are encouraged.

SUBSCRIBE AND SUBMIT

For more information on paper submission, featured articles, calls for papers, and subscription links visit:

www.computer.org/tsusc



IEEE COMPUTER SOCIETY

IEEE ComSoc
IEEE Communications Society

CEDA
IEEE Council on Electronic Design Automation

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

1 May

- HOST (IEEE Int'l Symposium on Hardware Oriented Security and Trust) San Jose, USA

7 May

- FCCM (IEEE Int'l Symposium on Field-Programmable Custom Computing Machines) Los Angeles, USA

14 May

- ICSE (IEEE/ACM Int'l Conference on Software Engineering) Melbourne, Australia

15 May

- IPDPS (IEEE Int'l Parallel and Distributed Processing Symposium) St. Petersburg, USA

17 May

- MOST (IEEE Int'l Conf. on Mobility, Operations, Services and Technologies) Detroit, USA

21 May

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

22 May

- ISMVL (IEEE Int'l Symposium on Multiple-Valued Logic) Matsue, Japan

29 May

- SERA (IEEE/ACIS Int'l Conf. on Software Engineering Research, Management and Applications) Orlando, USA

JUNE

7 June

- CAI (IEEE Conf. on Artificial Intelligence) Santa Clara, USA

10 June

- ICHI (IEEE Int'l Conf. on Healthcare Informatics) Houston, USA

12 June

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

17 June

- CVPR (IEEE/CVF Conf. on Computer Vision and Pattern Recognition) Virtual Conf.

21 June

- CHASE (IEEE/ACM Conf. on Connected Health: Applications, Systems and Eng Technologies) Orlando, USA

22 Jun

- CBMS (IEEE Int'l Symposium on Computer-Based Medical Systems) L'Aquila, Italy

23 June

- ICIS (IEEE/ACIS Int'l Conf. on Computer and Information Science) Wuxi, China

24 June

- LICS (ACM/IEEE Symposium on Logic in Computer Science) Boston, USA

26 June

- BSC (Biennial Symposium on Communications) Montreal, Canada
- COMPSAC (IEEE Computers, Software, and Applications Conf.) Torino, Italy
- JCDL (ACM/IEEE Joint Conf. on Digital Libraries) Santa Fe, USA

28 June

- WETICE (IEEE Int'l Conf. on Enabling Technologies: Infrastructure for Collaborative Enterprises) Paris, France

JULY

2 July

- SERVICES (IEEE World Congress on Services), Chicago, USA

3 July

- EuroS&P (IEEE European Symposium on Security and Privacy) Delft, Netherlands

10 July

- CSF (IEEE Computer Security Foundations Symposium) Dubrovnik, Croatia
- ICALT (IEEE Int'l Conf. on Advanced Learning Technologies) Orem, USA
- ICME (IEEE Int'l Conf. on Multimedia and Expo) Brisbane, Australia

 2469-7087/23 © 2023 IEEE

18 July

- ICDCS (IEEE Int'l Conf. on Distributed Computing Systems) Hong Kong
- SCC (IEEE Space Computing Conf.) Pasadena, USA
- SMC-IT (IEEE Int'l Conf. on Space Mission Challenges for Information Technology) Pasadena, USA

19 July

- ASAP (IEEE Int'l Conf. on Application-specific Systems, Architectures and Processors) Porto, Portugal

28 July

- ICCP (IEEE Int'l Conf. on Computational Photography), Madison, Wisconsin, USA

AUGUST

1 August

- IRI (IEEE Int'l Conf. on Information Reuse and Integration for Data Science), Bellevue, WA, USA

SEPTEMBER

4 September

- ARITH (IEEE Int'l Symposium on Computer Arithmetic), Portland, Oregon, USA
- RE (IEEE Int'l Requirements Eng. Conf.), Hannover, Germany

16 September

- SmartCloud (IEEE Int'l Conf. on Smart Cloud), Tokyo, Japan

25 September

- ACSOS (IEEE Int'l Conf. on Autonomic Computing and Self-Organizing Systems), Toronto, Canada
- MASS (IEEE Int'l Conf. on Mobile Ad-Hoc and Smart Systems), Toronto, Canada

OCTOBER

2 October

- ICCV (IEEE/CVF Int'l Conf. on Computer Vision), Paris, France
- LCN (IEEE Conf. on Local Computer Networks), Daytona Beach, USA

9 October

- e-Science (IEEE Int'l Conf. on e-Science), Limassol, Cyprus

18 October

- SecDev (IEEE Secure Development Conf.), Atlanta, USA

22 October

- VIS (IEEE Visualization and Visual Analytics), Melbourne, Australia

31 October

- CLUSTER (IEEE Int'l Conf. on Cluster Computing), Santa Fe, USA

NOVEMBER

4 November

- ICEBE (IEEE Int'l Conf. on E-Business Eng.), Sydney, Australia

12 November

- SC (Int'l Conf. for High-Performance Computing, Networking, Storage, and Analysis), Denver, USA

DECEMBER

1 December

- ICDM (IEEE Int'l Conf. on Data Mining), Shanghai, China

5 December

- BIBM (IEEE Int'l Conf. on Bioinformatics and Biomedicine), Istanbul, Turkey
- RTSS (IEEE Real-Time Systems Symposium), Taipei, Taiwan

11 December

- ISM (IEEE Int'l Symposium on Multimedia), Laguna Hills, USA

14 December

- BCD (IEEE/ACIS Int'l Conf. on Big Data, Cloud Computing, and Data Science Eng.), Ho Chi Minh City, Vietnam

18 December

- iSES (IEEE Int'l Symposium on Smart Electronic Systems), Ahmedabad, India