

COMPUTING edge

- Digital Health
- Edge Computing
- Smart Manufacturing
- Careers



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

Submit a paper today to the premier new open access journal in computing and information technology.

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

Submit your paper today!

Visit www.computer.org/oj to learn more.



STAFF

Editor
Cathy Martin

Publications Portfolio Managers
Carrie Clark, Kimberly Sperka

Senior Advertising Coordinator
Debbie Sims

Production & Design Artist
Carmen Flores-Garvey

Publisher
Robin Baldwin

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.

Editorial: Unless otherwise stated, bylined articles, as well as product and service descriptions, reflect the author's or firm's opinion. Inclusion in *ComputingEdge* does not necessarily constitute endorsement by the IEEE or the Computer Society. All submissions are subject to editing for style, clarity, and space.

Reuse Rights and Reprint Permissions: Educational or personal use of this material is permitted without fee, provided such use: 1) is not made for profit; 2) includes this notice and a full citation to the original work on the first page of the copy; and 3) does not imply IEEE endorsement of any third-party products or services. Authors and their companies are permitted to post the accepted version of IEEE-copyrighted material on their own Web servers without permission, provided that the IEEE copyright notice and a full citation to the original work appear on the first screen of the posted copy. An accepted manuscript is a version which has been revised by the author to incorporate review suggestions, but not the published version with copy-editing, proofreading, and formatting added by IEEE. For more information, please go to: http://www.ieee.org/publications_standards/publications

/rights/paperversionpolicy.html. Permission to reprint/republish this material for commercial, advertising, or promotional purposes or for creating new collective works for resale or redistribution must be obtained from IEEE by writing to the IEEE Intellectual Property Rights Office, 445 Hoes Lane, Piscataway, NJ 08854-4141 or pubs-permissions@ieee.org. Copyright © 2022 IEEE. All rights reserved.

Abstracting and Library Use: Abstracting is permitted with credit to the source. Libraries are permitted to photocopy for private use of patrons, provided the per-copy fee indicated in the code at the bottom of the first page is paid through the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923.

Unsubscribe: If you no longer wish to receive this *ComputingEdge* mailing, please email IEEE Computer Society Customer Service at help@computer.org and type "unsubscribe *ComputingEdge*" in your subject line.

IEEE prohibits discrimination, harassment, and bullying. For more information, visit www.ieee.org/web/aboutus/whatis/policies/p9-26.html.

IEEE Computer Society Magazine Editors in Chief

Computer

Jeff Voas, *NIST*

Computing in Science & Engineering

Lorena A. Barba, *George Washington University*

IEEE Annals of the History of Computing

Gerardo Con Diaz, *University of California, Davis*

IEEE Computer Graphics and Applications

Torsten Möller, *Universität Wien*

IEEE Intelligent Systems

Longbing Cao, *University of Technology Sydney*

IEEE Internet Computing

George Pallis, *University of Cyprus*

IEEE Micro

Lizy Kurian John, *University of Texas at Austin*

IEEE MultiMedia

Shu-Ching Chen, *Florida International University*

IEEE Pervasive Computing

Marc Langheinrich, *Università della Svizzera italiana*

IEEE Security & Privacy

Sean Peisert, *Lawrence Berkeley National Laboratory and University of California, Davis*

IEEE Software

Ipek Ozkaya, *Software Engineering Institute*

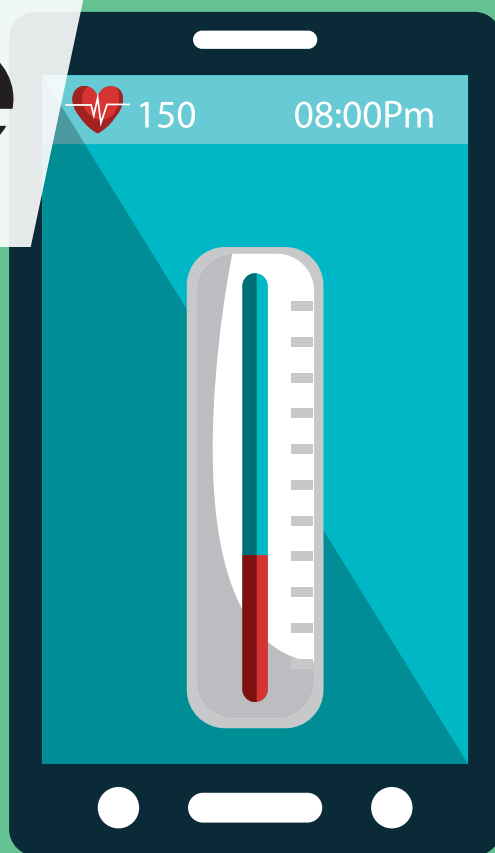
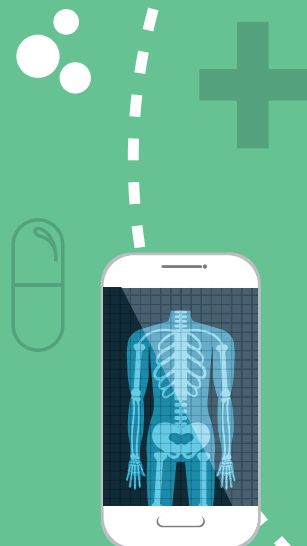
IT Professional

Irena Bojanova, *NIST*

JUNE 2022 • VOLUME 8 • NUMBER 6

COMPUTING

edge



8

Visual Analytics
of Smartphone-
Sensed Human
Behavior
and Health

28

Edge Artificial
Intelligence
Chips for the
Cyberphysical
Systems Era

39

Reversible
Execution for
Robustness
in Embodied AI
and Industrial
Robots



Digital Health

8 Visual Analytics of Smartphone-Sensed Human Behavior and Health

HAMID MANSOOR, WALTER GERYCH, ABDULAZIZ ALAJAJI, LUKE BUQUICCHIO, KAVIN CHANDRASEKARAN, EMMANUEL AGU, AND ELKE A. RUNDENSTEINER

18 Common Shortcomings in Applying User-Centered Design for Digital Health

LORRAINE R. BUIS AND JINA HUH-YOO

Edge Computing

24 Security and Privacy for Edge Artificial Intelligence

JAMES BRET MICHAEL

28 Edge Artificial Intelligence Chips for the Cyberphysical Systems Era

HIROSHI FUKETA AND KUNIO UCHIYAMA

Smart Manufacturing

34 Cognitive Digital Twins for Smart Manufacturing

MUHAMMAD INTIZAR ALI, PANKESH PATEL, JOHN G. BRESLIN, RAMY HARIK, AND AMIT SHETH

39 Reversible Execution for Robustness in Embodied AI and Industrial Robots

IVAN LANESE, ULRIC P. SCHULTZ, AND IREK ULIDOWSKI

Careers

45 A Brief History of Warehouse-Scale Computing

LUIZ ANDRÉ BARROSO

51 Software Engineering: A Profession in Waiting

DAVID LORGE PARNAS

Departments

4 Magazine Roundup

7 Editor's Note: Digitizing Healthcare

55 Conference Calendar

Subscribe to *ComputingEdge* for free at
www.computer.org/computingedge.



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

Discovering Opioid Use Patterns From Social Media for Relapse Prevention

The United States is experiencing an unprecedented opioid crisis. Through a multidisciplinary analytic perspective, the authors of this article from the February 2022 issue of *Computer* characterize opioid addiction behavior patterns by analyzing opioid groups from Reddit.com—including modeling online discussion topics, analyzing text co-occurrence and correlations, and identifying the emotional states of people with opioid use disorder.

Computing

Discovering Geometry in Data Arrays

Modern technologies produce a deluge of complicated data. In neuroscience, minimally invasive experimental methods can take recordings of large populations of neurons at high resolution under a multitude of conditions. Such data arrays possess nontrivial interdependencies along each of their axes. Insights into these data

arrays may lay the foundations of advanced treatments for nervous system disorders. The potential impacts of such data, however, will not be fully realized unless the techniques for analyzing them keep pace. Specifically, there is a need for methods for estimating the low-dimensional structure and geometry in big and noisy data arrays. This article from the November/December 2021 issue of *Computing in Science & Engineering* reviews a framework for identifying complicated underlying patterns in such data and recounts the key role that the Department of Energy Computational Sciences Graduate Fellowship played in setting the stage for this work.

IEEE Annals

of the History of Computing

The Work of Writing Programs: Logic and Inscriptive Practice in the History of Computing

This article from the October–December 2021 issue of *IEEE Annals of the History of Computing* explores the entanglement of logic and computing by focusing on the activity of writing. Although mathematical logic is sometimes cast

as the immaterial spirit of the computer's material body, the study of logic also takes place in the physical world through the manipulation of symbols on paper. Already in the 19th century, mathematical logic was understood to be related to mechanization, though not as the science behind an as-yet-uninvented technology. Rather, symbolic notations were seen as tools that opened possibilities but required new kinds of work. Turning to early electronic computing in the 1950s, the author observes that researchers similarly relied on novel inscriptive techniques to mitigate labor.

IEEE Computer Graphics

AND APPLICATIONS

Is the Perceived Comfort With CG Characters Increasing With Their Novelty?

Realistic characters from movies and games can cause strangeness and involuntary feelings in viewers, an effect known as the uncanny valley (UV). This article from the January/February 2022 issue of *IEEE Computer Graphics and Applications* revisits the central UV hypothesis, proposed by Masahiro Mori in 1970, to evaluate



its impact on people's perception of characters created using computer graphics (CG). The authors ask the following questions: 1) Are people feeling more comfortable with more recent CG characters than older ones? 2) Does charisma or familiarity with virtual humans correlate with perceived comfort? To answer these questions, the authors first replicated an experiment from 2012 and compared the perception concerning CG characters then and now, and then included images of more recent CG characters in the analysis. Results indicate that the perceived comfort increased over time when comparing the characters of 2012 and 2020.

IEEE Intelligent Systems

Embedding-Augmented Generalized Matrix Factorization for Recommendation With Implicit Feedback

Learning effective representations of users and items is crucially important to recommendation with implicit feedback. Matrix factorization derives representations of users and items by decomposing the given interaction matrix. However, existing matrix factorization-based approaches share the limitation that the interaction between user embedding and

item embedding is only weakly enforced by fitting the given individual rating value, which may lose potentially useful information. In this article from the November/December 2021 issue of *IEEE Intelligent Systems*, the authors propose a novel augmented generalized matrix factorization approach that can incorporate the historical interaction information of users and items for learning effective representations. Despite the simplicity of the proposed approach, extensive experiments on four public implicit feedback datasets demonstrate that it outperforms state-of-the-art counterparts.

IEEE Internet Computing

Quantum Software as a Service Through a Quantum API Gateway

As quantum computers mature, the complexity of quantum software increases. As we move from the initial standalone quantum algorithms toward complex solutions combining quantum algorithms with traditional software, new software engineering methods and abstractions are needed. Nowadays, quantum computers are usually offered in the cloud, under a pay-per-use model, leading to the adoption of the service-oriented good practices that dominate the cloud today. However, specific

adaptations are needed to reap the benefits of service-oriented computing while dealing with quantum hardware limitations. In this article from the January/February 2022 issue of *IEEE Internet Computing*, the authors propose the Quantum API Gateway—an adaptation of the API Gateway pattern that considers the fact that quantum services cannot be deployed as traditional services.

IEEE micro

Artificial Intelligence Best Practices in Smart Agriculture

Smart agriculture, with the aid of artificial intelligence (AI), is playing a pivotal role to ensure agriculture sustainability. AI techniques are employed in soil and irrigation management, weather forecasting, plant growth, disease prediction, and livestock management. The authors of this article from the January/February 2022 issue of *IEEE Micro* review recent AI techniques that have been deployed in these domains.

IEEE MultiMedia

Are Remote Play Streaming Systems Doomed to Fail? A Network Perspective

Digital games represent one of the most compelling fields in computer

science, embodying a wide variety of technical challenges. Thanks to the evolution of streaming and broadband technology, new service provisioning schemes have emerged. Remote play streaming services represent an interesting case study deserving a thorough investigation. To this end, the authors of this article from the October–December 2021 issue of *IEEE MultiMedia* present a network measurement study that can be useful to create traffic models and help researchers identify issues, guiding architecture, and protocol design. Moving beyond latency and jitter issues, the purpose is to understand whether remote play streaming services can operate through regular connectivity or are doomed to fail as happened to some pioneer providers. The authors deploy a testbed to test the impact of network limitations and emphasize the role of the available bandwidth in this context.



Predicting Job Performance Using Mobile Sensing

The authors of this article from the October–December 2021 issue of *IEEE Pervasive Computing* hypothesize that behavioral patterns of people are reflected in how they interact with their mobile devices and that continuous sensor data passively collected from their phones and wearables can infer their job performance. The authors study day-to-day job performance (improvement, no change, decline)

of 298 information workers using mobile sensing data and offer data-driven insights into what data patterns may lead to a high-performing day. Through analyzing workers' mobile sensing data, the authors predict their performance on a handful of job performance questionnaires with an F-1 score of 75%. In addition, through numerical analysis of the model, they gain insights into how individuals must change their behavior so that the model predicts improvements in their job performance.



Personal IoT Privacy Control at the Edge

This article from the January/February 2022 issue of *IEEE Security & Privacy* introduces a privacy manager for Internet-of-Things data based on edge computing. This poses the advantage that privacy is enforced before data leaves the control of the user, who is provided with a tool to express data-sharing preferences based on context-aware privacy language.



Toward Autonomic, Software-Intensive Digital Twin Systems

Digital twins (DTs) mirror and model the characteristics and properties of dynamic, real-world entities known as real twins (RTs). Ensuring the delivery of consistent and reliable RT insights over time demands that DTs preserve the correspondence

with their counterparts, notwithstanding change. Read more in this article from the March/April 2022 issue of *IEEE Software*.



Intelligent Traffic Signal Automation Based on Computer Vision Techniques Using Deep Learning

Traffic congestion in highly populated urban areas is a huge problem these days. Researchers have proposed many systems to monitor traffic flow and handle congestion through different techniques. But the current systems are not reliable enough to perceive traffic signals in real time. The authors of this article from the January/February 2022 issue of *IT Professional* aim to build a system that can efficiently perform real-time environments to solve the traffic congestion problem through signal automation. Since vehicle detection and counting are crucial in any traffic system, the authors use state-of-the-art deep-learning techniques to detect and count vehicles in real time. They then automate the signal timings by comparing the count of traffic on all sides of a junction. These automated signal timings sufficiently reduce congestion and improve traffic flow. 🚦

Join the IEEE
Computer Society
computer.org/join



Editor's Note

Digitizing Healthcare

Digital transformation of the healthcare industry is leading to enhanced efficiency, personalization, and precision—with the ultimate goal of improving patient experiences and outcomes. From communication portals and wearable devices to complex data analytics, digital health is having a big impact on patients, providers, and researchers. This *ComputingEdge* issue explores the latest innovations and obstacles in digital health.

"Visual Analytics of Smartphone-Sensed Human Behavior and Health," from *IEEE Computer Graphics and Applications*, describes the emerging field of interactive visual analytics as a way to facilitate the discovery and correction of user-provided ground-truth health data. "Common Shortcomings in Applying User-Centered Design for Digital Health," from *IEEE Pervasive*

Computing, discusses challenges that designers often face when developing technologies for the healthcare space.





Another game-changing technology being implemented today is edge intelligence, or edge computing that incorporates AI. *IEEE Security & Privacy's* "Security and Privacy for Edge Artificial Intelligence" explains the concept and details its security-related benefits and disadvantages. *Computer's* "Edge Artificial Intelligence Chips for the Cyber-physical Systems Era" reports on the microprocessor architecture that enables energy-efficient edge AI for applications such as autonomous driving and factory automation.

Smart manufacturing is also incorporating AI in creative new ways. In *IEEE Intelligent Systems' "Cognitive Digital Twins for Smart Manufacturing,"* the

authors illustrate how digital twins with AI capabilities are adding value in Industry 4.0. In *IT Professional's "Reversible Execution for Robustness in Embodied AI and Industrial Robots,"* the authors combine traditional AI planning with reversibility and embodied AI when programming industrial robots for assembly operations.

Finally, this *ComputingEdge* issue features two articles on computing careers. In *IEEE Micro's "A Brief History of Warehouse-Scale Computing,"* the 2020 IEEE Computer Society/ACM Eckert-Mauchly Award winner recounts his professional journey that led to his breakthrough work at Google. In *Computer's "Software Engineering: A Profession in Waiting,"* the author argues that we need licensing to transform software development into an engineering discipline. 🤖

Visual Analytics of Smartphone-Sensed Human Behavior and Health

Hamid Mansoor , Walter Gerych, Abdulaziz Alajaji , Luke Buquicchio , Kavin Chandrasekaran ,
Emmanuel Agu , and Elke A. Rundensteiner, Worcester Polytechnic Institute, Worcester, MA, 01609, USA

Smartphone health sensing tools, which analyze passively gathered human behavior data, can provide clinicians with a longitudinal view of their patients' ailments in natural settings. In this Visualization Viewpoints article, we postulate that interactive visual analytics (IVA) can assist data scientists during the development of such tools by facilitating the discovery and correction of wrong or missing user-provided ground-truth health annotations. IVA can also assist clinicians in making sense of their patients' behaviors by providing additional contextual and semantic information. We review the current state-of-the-art, outline unique challenges, and illustrate our viewpoints using our work as well as those of other researchers. Finally, we articulate open challenges in this exciting and emerging field of research.

THE CURRENT HEALTHCARE SYSTEM is under-resourced and schedule-driven with patients receiving little care outside of appointments. Consequently, patient assessments are infrequent, typically months apart, and often result in late diagnoses that worsen their prognoses. Emerging research is exploring the use of sensor-rich smartphones that are now owned by over 80% of the U.S. population,[†] to passively detect various ailments, and continuously gather valuable health behavior information and corresponding contexts. Ailments such as depression^{5,17} and influenza⁹ can be detected early by analyzing sensor data collected from smartphones using machine learning models. This novel paradigm is called *smartphone health sensing* or *smartphone ailment phenotyping*. Early detection can significantly improve health outcomes.[‡] Passive smartphone phenotyping provides clinicians with an objective, contextualized

picture of their patients' lives in the real world. Such evidence can then be used to support treatment decisions as patient self-reports may be inaccurate due to recall bias and exaggeration. However, analysis of real world smartphone-sensed health data is challenging.

In addition to traditional issues such as the highly multivariate and complex nature of such spatio-temporal data, smartphone-sensed data analysis faces unique challenges such as the need to disambiguate noisy ground truth labels of health behaviors and context in natural settings. While passive data gathering in natural settings yields realistic data, it also means that users often provide wrong or no labels when they are busy with their lives. Such labeling issues in turn lead to weak supervision for machine learning modeling. Smartphone health inference and phenotyping also faces unique challenges as user behaviors indicative of health status are often intertwined with other unrelated real world activities. Ultimately, the smartphone user's specific situations, contexts, and health status at any point in time are not always clear. Moreover, multiple ailments can have the same smartphone signature or phenotype, leading to confounding effects. Finally, the degree to which users express each symptom of the same underlying ailment vary a lot, making intersubject comparisons challenging.

Prior work on IVA for health related data were typically on structured data from sources such as

[†]<https://www.pewresearch.org/internet/fact-sheet/mobile/>

[‡]<https://www.webmd.com/depression/guide/untreated-depression-effects>

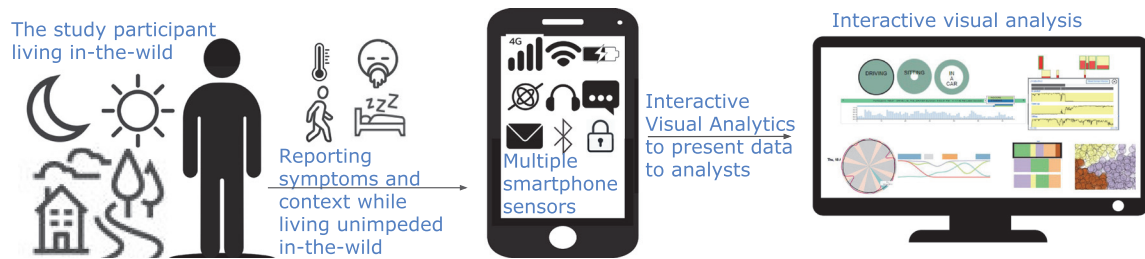


FIGURE 1. During user studies to gather labeled data for developing smartphone health inference models, a user lives in-the-wild while their smartphone passively gathers data continuously. Periodically, the user provides ground truth labels of their context (situation) and health symptoms on their phone. Users often provide wrong or no labels when they become busy with their lives, which presents a challenge for supervised machine learning methods. Visual analytics can assist data scientists in correcting wrong or missing labels, cleaning such data before machine learning modeling, and debugging of such models. For clinicians using the final machine learning-based passive health inference models, IVA provides additional patient context and semantic information on the health symptoms, and comparisons with other patients, which enhances interpretability and trust of models' outputs.

Electronic Health Records (EHRs) with clear definitions of data fields and coding of health related events and relevant patient information.^{6,15} IVA systems for sensor-based health data have visualized low level health and wellness variables such as meals, sleep patterns, and step counts, which were gathered using health wearable devices such as Fitbits.¹⁸ We focus on IVA systems for making higher level inferences [e.g., detecting Traumatic Brain Injury (TBI)] passively from these low level health variables and symptoms captured in-the-wild using smartphones (Figure 1).

A prior survey by Polack *et al.* reviewed IVA methods to analyze *Mobile Health (mHealth)* data in general.¹⁵ Topics covered included the visual representation of complex and multivariate temporal data, interactive cohort selection, and trend mining. Our viewpoints are more focused on specific IVA support for passive smartphone health sensing or phenotyping, which typically utilize machine learning. We add to this exciting and emerging field by distilling novel viewpoints, defining key stakeholders, outlining unique challenges arising from this new method of monitoring health, and providing concrete examples to illustrate how IVA methods can provide additional insights and assist specific stakeholders. Our viewpoints focus on the unique challenges that stem from the weakly and sometimes incorrectly supervised nature of *in-the-wild* smartphone health sensing studies and the dynamic, ambiguous, and sometimes confounding behaviors of the monitored user. Aspects our viewpoints cover include support for correcting user-provided annotations, enhancing the understanding, contextualization, and sensemaking of smartphone-sensed health behaviors, and population-level visualizations and subject health status comparisons.

SMARTPHONE HEALTH SENSING—STAKEHOLDERS AND CHALLENGES

There are generally two groups of stakeholders for smartphone health sensing or phenotyping: 1) *Clinicians and Health Professionals* who seek to use such systems to understand their patients' ailment-related real world behaviors and symptom trajectories in the real world better, and identify concerning behaviors early; 2) *Data Scientists* who develop computational and machine learning models to passively detect ailments using weakly supervised smartphone-sensed data with noisy labels. Their ultimate goal is to deploy those models to continuously assess and monitor the patient and detect ailments early in a completely unsupervised fashion. These two groups of stakeholders face different but related challenges, which we now summarize.

Health Behavior Understanding Challenges

While inferences about health and contexts made using machine learning on smartphone data can be accurate,^{16,17} they are typically not *explainable* nor do they incorporate expert knowledge. For example, while machine learning can detect sleep duration and quality accurately using smartphone data,¹ it does not capture a comprehensive picture of potential causes of sleep disruptions which are important for healthcare professionals. Such disruptions may be *explainable* if additional *human understandable* information and the occurrence of comorbidities such as participants' increased reported stress were *visually* linked to such

inferences. Clinicians who use smartphone health sensing systems to monitor their patients as well as data scientists who develop them both face health behavior understanding challenges.

Symptom and Health Behavior Contextualization Challenges

Human life involves multiple, intertwined experiences. Clinician end users as well as data scientist model developers typically would like temporal information regarding their subjects' situation and trajectory leading up to a smartphone-sensed assessment, concurrently and afterward in order to fully contextualize it. Health analysts can utilize *visual* methods to contextualize such information¹⁵ to disambiguate confounding scenarios and improve the specificity of diagnoses. For instance, while reductions in a smartphone user's step count may be caused by depression, it may also be a short-term reduction caused by fatigue because the user engaged in strenuous exercise the previous day.

Smartphone Data Labeling Challenges

Data scientists creating smartphone health assessment models using machine learning require labeled, *real world* datasets. Typically, an app installed on users' smartphones, continuously gathers sensor data as they live their lives. To provide ground truth labels, users periodically respond to questions to report their health condition¹⁷ symptoms, as well as corresponding contexts visited, activities performed, and social situations experienced.¹⁶ Label data collection studies can be disruptive leading to two data science issues: 1) *Missing Labels*: participants fail to provide health or context labels when they are busy or distracted. Participant response rates to ground truth questions also vary leading to imbalanced datasets, and 2) *Wrong Labels*: participants make human errors in providing labels due to carelessness or recall bias.¹²

OUR POSTULATIONS ON HOW IVA CAN ENHANCE SMARTPHONE HEALTH SENSING

Interactive data visualizations are useful for analyzing multivariate data.^{7,14} Polack *et al.*¹⁵ previously highlighted some research directions for IVA for mHealth data broadly including visualizing its multi-scale, temporal nature. We build on and extend this work by summarizing specific *Viewpoints* on how IVA can be useful for the different stakeholders of

smartphone health sensing and phenotyping, and present concrete, illustrative examples.

IVA Support for Data Scientists

V1: Visual support for detecting mislabeled data to improve the model development. Traditionally, IVA works for data cleaning in complex domains such as multimedia, trajectory, and textual data,⁸ rely on metrics such as anomaly scores to alert users of poor quality data. In-the-wild gathered smartphone-sensed data is more multicontextual in nature, which makes it difficult to rely solely on such computational methods to discover poorly labeled data. IVA is well suited to present the complex characteristics of smartphone-sensed health data¹⁵ and can leverage multiple visual metaphors to display automatically derived metrics and other intuitive cues for anomalous data indicative of mislabeling effectively. For instance, activities labeled as occurring simultaneously but which are unlikely to truly be co-occurring (e.g., standing while driving) can be visually highlighted to make them easy to discover.

Context Mislabel EXplorer (COMEX)¹¹ is an IVA framework that facilitates the discovery of mislabeled smartphone sensed data by incorporating visuals that provide additional context. COMEX analyzes continuously gathered, in-the-wild smartphone data¹⁶ with participant-provided ground truth health and context labels. COMEX combines the visualization of computed label anomaly scores with visual metaphors designed to address the multicontextual and continuous nature of real world, labeled smartphone sensor data. For instance, COMEX highlights unlikely co-occurrence of activities as a clue for discovering wrong labels [e.g., "Driving" while "Indoors" Figure 2(A) and (B)], alerting analysts about potential mislabels. Unlikely context and activity durations can also provide another clue for wrong user labels [e.g., "Walking" for 15 h, Figure 2(C)]. However, users tend to label data at frequencies that vary over time, providing more labels when they are free and less when busy. Consequently, detecting the start-end and continuation of contexts over time can be confusing. COMEX deals with temporal variations in labeling frequency by presenting visual indicators ("chunks") of contextual continuity to make them easy to identify continuing contexts, compare user-reported context durations, and identify likely wrong reports. The intuitive use of simple metaphors to highlight suspicious context co-occurrence and duration demonstrate that IVA is well-suited to assist in label correction of smartphone sensed data.

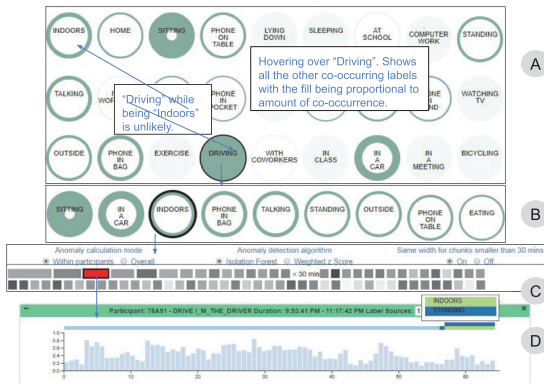


FIGURE 2. COMEX. (A) Showing labels across participants, ordered by occurrence. Hovering over a circle shows its most commonly co-occurring labels with the fill being proportional to co-occurrence. (B) Clicking on a circle shows Chunks of the selected label with length encoding time. The chunks are ordered by duration and their opacity encodes their average anomaly score. Clicking on a chunk shows its details as a histogram. The bars above the histogram are the co-occurring labels for the sessions.

V2: Visual support for labeling unlabeled data to improve the model development. In order to discern the likely labels of unlabeled data, some IVA methods integrate contextual visual cues of multivariate data to enable analysts to assign labels to data more confidently.³ However, such approaches require data with verifiable ground truth labels, which is not the case in

in-the-wild smartphone data, where the user may not have provided labels and exact ground truth labels are not known. In cases where smartphone data are partially labeled, a semisupervised visual paradigm can be utilized, wherein the probable labels of unlabeled data are discerned from their visual similarity to labeled data. To make labeling easy, IVA tools can provide contextual details from passively sensed data and highlight similarities between labeled and unlabeled data. For instance, time periods during which the smartphone logs show calls have been received as well as high noise levels can be labeled as “conversation” based on similarity between features corresponding to this time period and other instances of data labeled as “conversation.”

Detecting Erroneous Labels using Feature-linking Insights (DELFI) (see Figure 3)¹² is an IVA framework to highlight unlabeled data, suggest similarities to labeled data instances in terms of *sensor feature values*, and facilitate the assignment of labels with confidence. DELFI utilized a Multi-Feature Similarity Linking paradigm (inspired by Nguyen *et al.*¹⁴) to visually link feature-similar data with an overlay of contextual information, enhancing intuition. Interacting with continuous “chunks” of labeled or unlabeled data shows the most feature-similar chunks (based on Euclidean distance between feature values) for visual linking [see Figure 3(A)], to discover potentially mislabeled data and assign labels to unlabeled data. The values of soft sensors such as the apps running, call, and SMS logs, and charging status add another layer of

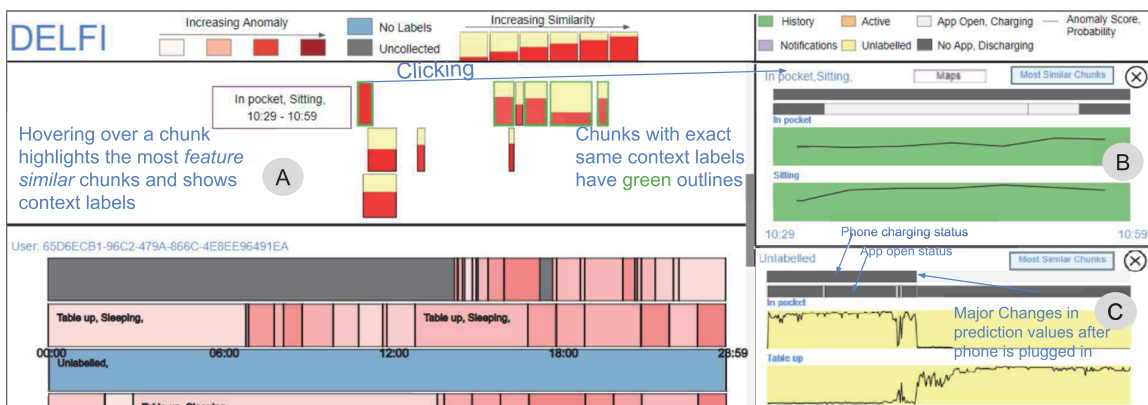


FIGURE 3. DELFI: (A) *Habit View*: Shows every user's participation days as Continuous Context Chunks. Hovering over a chunk hides all others except those that are most feature similar. (B) *Chunk Detail View*: Shows chunk details like the labeling mechanism (these labels were provided using “History”). The labels making up the context are split as bars and the lines show the respective anomaly scores or probability values (for unlabeled data) for the comprising data. The two gray bars represent the charging status (first bar) and app usage status (second bar) for the data sessions. (C) Unlabeled chunk. The labels with the highest average probability values are shown and the lines in the bars represent probability values for individual sessions.

explainability. For instance, around the time the phone was plugged in, the unlabeled chunk in Figure 3(C) had an increased likelihood of having the label “Table up” (phone position) and a decline for the label “In pocket.” Visual overlays increase the analyst’s confidence by presenting intuitive contextual information to enhance interpretation. Such intuition is difficult to generate with nonvisual and purely computational methods.

V3: *Visual clustering of smartphone-sensed data* to improve scalability and steer the development of machine learning classifiers for health applications. Groups of data similar in terms of sensor feature values can be visualized with overlays of analyst-understandable semantic information that can help in building classification models. For instance, grouping days with poor sleep reports can help identify user phenotypes and patterns in smartphone-sensed data features that can be predictive of them. In addition, as the scale of such studies grows, *Population level* analysis becomes necessary for understanding differences between groups and subpopulations of people. IVA is well-suited for the task of cohort selection and comparison as noted by Polack *et al.*¹⁵ For instance sleep problems in late shift workers versus early shift workers can be contextualized based on the distribution of sensor-detected sleep hours, and can help analysts in assigning chronotypes (groups with similar sleep-wake cycles)¹ (e.g., morning person versus night owl) to participants. Presenting inferred as well as reported symptom data with contextual information such as phone interactions, mobility patterns, and days of the week can enable analysts to understand the differences in sensed data between participants and allow for larger scale, longer term analyses.

Prior work enabled unsupervised analysis of multi-feature data by presenting results from multiple clustering and dimension reduction algorithms.⁷ We are researching and developing INTERactive Observation of Smartphone Inferred Symptoms (INTOSIS), an IVA tool which adopted a similar approach for a large-scale smartphone-sensed dataset in an ongoing study. The data include sensors such as anonymized geolocation and activity levels along with sparse daily and weekly symptom labels. A ranked list of clustering results (based on quality of clustering) was generated using various algorithms such as K-Means and spectral distancing [see Figure 6(E)] from clustering all days across users based on sensor features. The clusters are then visualized across a plane using *t*-distributed stochastic neighbor embedding (*t*-SNE), a visual dimension reduction technique. This enables analysts to see similarity of features between symptomatic

days and provide explanations for the distributions of clusters. In addition, this shows the sensor-detected factors that may be indicative of symptoms. For example, a cluster with several days labeled as “Poor sleep” may be correlated with less time at the participant’s primary location at night.

The “Feature Averages Heatmap” [see Figure 6(D)] shows the feature value distribution, ordered by their importance (ANOVA F-statistic). This shows the defining characteristics of clusters [e.g., purple cluster shows days with more than usual time spent at home, Figure 6(D)] and assign semantic meaning to objective sensor data. IVA enables easier interpretation of such unsupervised data in a way that lets analysts make such important associations.

IVA Support for Health Professionals

V4: *Visualizing anomalous, passively sensed unhealthy patterns of user behaviors.* Deviations from routines can provide health analysts with clues about causes of symptoms/concerning behaviors. Human Bio behavioral Rhythms (HBR) such as sleep-wake cycles or circadian rhythms and their disruptions have health ramifications and are detectable from smartphone data.¹ While such computational approaches are accurate, they usually provide little *explainability*, which is an issue as the scope of such studies grows and as users provide more sparse labels with increased study durations. Alternate nonvisual analysis methods exploit the multiscale temporality of behavioral rhythms derived from smartphone-sensed data. IVA techniques can combine multiple views that facilitate contextualization of multiscale temporal data for trend and pattern mining, which can reveal important, health-relevant information¹⁵ such as sleep patterns and stress levels during hectic times such as weekdays versus more relaxed times such as weekends, etc.

ARGUS¹⁰ is an IVA framework that displays smartphone-sensed HBRs and disruptions in them, using multiple visual concepts to assist analysts in not only identifying but also explaining HBR disruptions. ARGUS utilizes a novel Rhythm Deviation Score (RDS) that quantifies the degree of periodicity of the underlying sleep wake-cycle based on sensor data. Each participant day is assigned an RDS score, which can then be visualized effectively in conjunction with other contextual information. ARGUS uses a glyph based on the Z-glyph,⁴ a visual metaphor to present disruptions from the norm. The black circle [see Figure 4(A)] represents the overall rhythm (larger circle means more rhythmic) and the dips in the purple line represent days with disruptions. Bigger dips toward the center

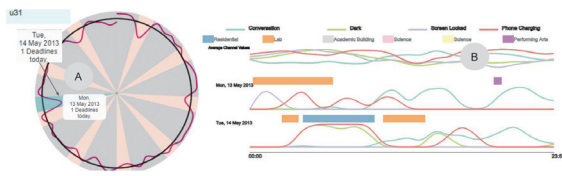


FIGURE 4. (A) Larger black circle means more rhythm. The purple line shows daily disruption in rhythm (closer to center means higher disruption). Every slice is a day, with the beige slices being weekends. Days can be selected for analysis in (B). (B) displays the daily distribution across sensor values (lines) along with durations spent in geoclusters (colored bars over lines are place types).

indicate bigger HBR disruptions. Contextual data about possible causes of HBR disruptions such as academic and project deadlines and the places visited are overlaid on the glyph. Presenting additional contextual information such as the type of places visited enables analysts to assign contextual meaning to the sensed HBR (see Figures 4 and 5).

For instance, a university student's¹⁷ smartphone-sensed rhythm was disrupted for two days by deadlines [see Figure 4(A)]. Exploration shows that they were in a "Lab" during the early hours of the 2 days [see Figure 4(B)]. Such detailed visual analysis not only shows concerning data, but also helps to *explain* and *contextualize* it. Nonvisual methods are limited in their ability to make use such connections in order to explain the degree of HBR periodicity.

V5: Visually overlay health markers and symptom reports to assign semantic values to objective sensor data for health analysis. Smartphone-sensed data are often anonymized, with loss of information that can potentially explain symptoms. For instance, not knowing semantic information about a participant's work-life routine may make it harder to explain depressive symptoms since staying longer at work has been linked with depression.^{2, 13} Consequently, the ability to *semantically* label unlabeled/anonymized smartphone-sensed data can be valuable (e.g., labeling the place where a participant "Stays" most as their home⁵). IVA can provide views to show temporal trends in smartphone-sensed data¹⁵ that can enable semantic labeling of objective sensor data. INTO-SIS utilizes the same large-scale dataset from the ongoing study previously mentioned, to visualize day-level mobility features, based on location data per day [daily values shown for a user in Figure 6(A) and (C)]. Figure 6 (B) shows a list of the geoclusters (clustered using DBSCAN), sorted by the duration of participants' "Stays" in them. For each geocluster, the average 24-h

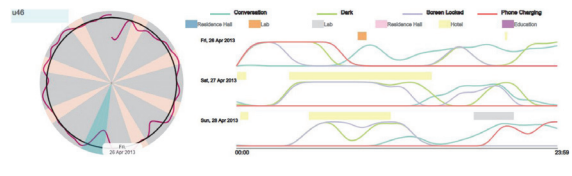


FIGURE 5. Contextualizing a rhythm break by showing human understandable information. For instance there was a large break in rhythm detected over a weekend where we see the participant in a "Hotel."

distribution of presence is shown (flatter lines indicate uniform presence). Days with positive symptom responses can be highlighted [see Figure 6(A)] for drill-down. For instance, the user was in geocluster 0 approximately every day [see Figure 6(B)], with uniform distribution, suggesting this is a residence. They are in geocluster 3, the second longest amount of time. This geocluster is likely their workplace, as they were never in that geocluster before 12 pm nor on weekends and also not since mid-March, when social distancing went into effect in the U.S. due to COVID-19 and workplaces were closed. Also, after mid-March, there was a decrease in location entropy, which measures how much the user visited popular locations. The ability to flexibly define and assign *semantic labels* can help analysts assign meaningful/ predictive labels to unlabeled data and improve inference.

CALL TO ACTION: OPEN RESEARCH CHALLENGES

While significant progress has been made toward realizing the vision of using IVA to enhance smartphone-sensed health assessment tools, several open challenges still need to be solved. As the scope of studies grows, data scientists and health experts will need to address challenges associated with increases in both the duration of studies and the number of users monitored including

- ▶ **Scalable visualizations:** As the scope of in-the-wild studies broadens and larger groups of smartphone users are analyzed, visual scalability must be considered. Specifically, the ability to visualize larger numbers of users with more data and over longer periods of time becomes important. The wealth of data available on the current COVID-19 pandemic is a case in point. While this may be tackled by using longer time windows (weeks instead of days) for analysis, this could inversely affect the quality or precision of the inferred results as ailments may not manifest

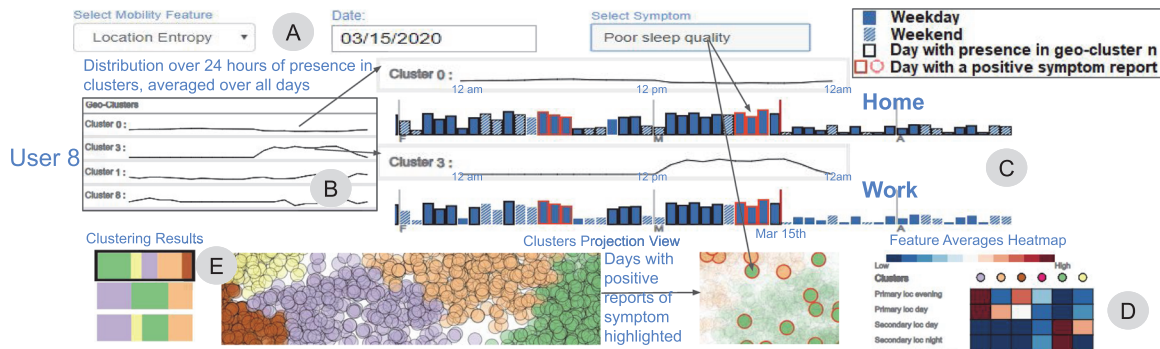


FIGURE 6. INTOSIS: Interacting with a geocluster distribution (B) shows the days where the user was in that cluster for some time with a black stroke. Flatter lines mean even presence for all hours of the day. Specifying a symptom (A) shows days with positive instances with a red stroke. Based on sensor features, data were analyzed using multiple clustering and dimension reduction techniques (E) to project each day (circle = day) of data across all participants on a 2-D plane. The clustering results are ordered by quality. Average feature value distribution across clusters is shown as a heatmap. Red and blue represent high and low values, respectively (D).

clearly on larger time scales. The visualization techniques themselves have to be scalable and interactive. While some of our work has begun to address this challenge, more research is required.¹⁵ Visual clustering can also be used along with multiscale visual interactive techniques such as zooming, filtering, and details on demand which can facilitate drill-downs, high-level meta views, and inter and intragroup analysis as the number of participants grows.

- › *Visualizing causative presymptom patterns:* To facilitate retrospective analysis and the discovery of behaviors that caused illness, it may be useful to support visual lookbacks on data and highlight users' behavior patterns that occurred frequently leading up to specific symptom reports. These can serve as clues for potential ailment causes. For instance, it may be informative to display a pattern of higher than usual activity levels on days preceding the time when a subject reported fatigue.
- › *Visual encodings that align with experts training and build on domain knowledge.* Many users of IVA tools for smartphone sensing such as psychiatrists, doctors, and nurses will already have substantial prior domain training that they could bring to bear in interpreting various visual clues. Working with experts to develop visual encodings that build on their domain knowledge is important. For instance, psychologists and health experts are trained that a patient's behavior differs on weekdays versus weekends. In our work with them, they liked encodings of user

behaviors on weekdays versus weekends along with specific smartphone sensed patterns such as mobility or app usage during day versus night. Such visuals will enable the experts to apply their domain expertise to the analysis task. Therefore, any data visualizations for smartphone data should explicitly encode contextual information accepted by the health domain.

- › *Multifacet aggregation of contextual variables.* A smartphone user's context has multiple facets including their activity, app being used and the type of place they are at. Displaying these disparate bits of information as separate visual streams across long periods of time can quickly become overwhelming. Future work may consider multicontext visual metaphors that represent analyst-specified aggregations of various sensor channels or those accepted by the health domain in a human-understandable fashion. For instance, aggregating low user activity, and environmental light and sounds levels as well as information that the user stayed at the same geolocation at nights to be visually displayed as "sleeping at night." Once labeled, subsequently collected data can then be automatically labeled with the "sleeping at night" label as appropriate.
- › *User-friendly visual metaphors for variable visualization literacy* among health experts and general public. Smartphone-sensed health detection is a multidisciplinary topic, including researchers with varying levels of visualization, computer, and data science literacy. Future

work can develop universally comprehensible visual metaphors to lower the learning curve for researchers across the spectrum of expertise in the health domain, who use IVA frameworks. Additionally, smartphone-sensed studies may also benefit from deploying on-device, user-friendly visualizations, and *gamification* strategies to increase participant compliance such as encouraging accurate and frequent labeling. Visualizing smartphone-sensed health data on a personal level may also mitigate the privacy issues that are inherent in most smartphone data gathering projects by giving participants access to their own data. Likewise, health recommendations generated by automated models can be provided to the user without analyst intervention.

CONCLUSION

Smartphones-sensed human behavior and health data are rapidly increasing in both quantity and complexity. Smartphone health sensing and phenotyping analyses try to utilize objective data to passively assess the health of the smartphone user and derive meaningful insights. However, labeling issues and the complex nature of smartphone-sensed, real world data present challenges to analysts who utilize predominately computational approaches. Interactive data visualizations are a viable alternative that can assist health professionals in such phenotyping analysis and improve the understanding and contextualization of health behaviors. IVA can also provide data scientists with valuable tools for mitigating data labeling issues and debugging machine learning health inference models during model development. In this visualization viewpoints article, we posited that interactive data visualization is an exciting approach to empower health scientists to discover smartphone-sensed human behaviors and symptoms that are predictive of health ailments and presented illustrative examples. 🌈

ACKNOWLEDGMENT

This material is based on research sponsored by DARPA under agreement number FA8750-18-2-0077. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as

necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government.

REFERENCES

1. S. Abdullah, E. L. Murnane, M. Matthews, and T. Choudhury, "Circadian computing: Sensing, modeling, and maintaining biological rhythms," in *Mobile Health*. New York, NY, USA: Springer, pp. 35–58, 2017.
2. P. Afonso, M. Fonseca, and J. F. Pires, "Impact of working hours on sleep and mental health," *Occup. Med.*, vol. 67, no. 5, pp. 377–382, 2017.
3. B. Jurgen *et al.*, "VIAL: A unified process for visual interactive labeling," *Vis. Comput.*, vol. 34, no. 9, pp. 1189–1207, 2018.
4. N. Cao, Y.-R. Lin, D. Gotz, and F. Du, "Z-glyph: Visualizing outliers in multivariate data," *Inf. Vis.*, vol. 17, no. 1, pp. 22–40, 2018, doi: 10.1177/1473871616686635.
5. W. Gerych, E. Agu, and E. Rundensteiner, "Classifying depression in imbalanced datasets using an autoencoder-based anomaly detection approach," in *Proc. IEEE 13th Int. Conf. Semantic Comput.*, 2019, pp. 124–127.
6. Z. Jin *et al.*, "Carepre: An intelligent clinical decision assistance system," *ACM Trans. Comput. Healthcare*, vol. 1, no. 1, pp. 1–20, 2020.
7. B. C. Kwon *et al.*, "Clustervision: Visual supervision of unsupervised clustering," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 142–151, Jan. 2018.
8. L. Shixia *et al.*, "Steering data quality with visual analytics: The complexity challenge," *Vis. Inform.*, vol. 2, no. 4, pp. 191–197, 2018.
9. A. Madan, M. Cebrian, S. Moturu, K. Farrahi, and A., "Sensing the" health state" of a community," *IEEE Pervasive Comput.*, vol. 11, no. 4, pp. 36–45, Oct./Dec. 2011.
10. H. Mansoor *et al.*, "AR- GUS: Interactive visual analytics framework for the discovery of disruptions in bio-behavioral rhythms," in A. Kerren, C. Garth, and G. E. Marai, eds., in *EuroVis 2020 - Short Papers*. Genève, Switzerland: The Eurographics Assoc., 2020, doi: 10.2312/evs.20201043.
11. H. Mansoor, W. Gerych, L. Buquicchio, K. Chandrasekaran, E. Agu, and E. Rundensteiner, "Comex: Identifying mislabeled human behavioral context data using visual analytics," in *Proc. IEEE 43rd Annu. Comput. Softw. Appl. Conf.*, vol. 2, pp. 233–238, 2019, doi: 10.1109/COMPSAC.2019.10212.

12. H. Mansoor, W. Gerych, L. Buquicchio, K. Chandrasekaran, E. Agu, and E. Rundensteiner, "DELF: Mislabelled human context detection using multi-feature similarity linking," in *Proc. IEEE Vis. Data Sci.*, 2019, pp. 11–19, doi: 10.1109/VDS48975.2019.8973382.
13. Martin, M., R. Weibel, C. Röcke, and S. M. Boker, "Semantic activity analytics for healthy aging: Challenges and opportunities," *IEEE Pervasive Comput.*, vol. 17, no. 3, pp. 73–77, Jul./Sep 2018.
14. P. H. Nguyen, C. Turkay, G. Andrienko, N. Andrienko, O. Thonnard, and J. Zouaoui, "Understanding user behaviour through action sequences: From the usual to the unusual," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 9, pp. 2838–2852, Sep. 2019.
15. P. Polack, M. Sharmin, K. de Barbaro, M. Kahng, S. Chen, and D. Chau, "Exploratory visual analytics of mobile health data: Sensemaking challenges and opportunities," in *Mobile Health*. Cham, Switzerland: Springer, 2017, pp. 349–360.
16. Y. Vaizman, K. Ellis, G. Lanckriet, and N. Weibel, "Ex- trasensory app: Data collection in-the-wild with rich user interface to self-report behavior," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2018, pp. 1–12.
17. R. Wang et al., "Studentlife: Assessing mental health, academic performance and behavioral trends of college students using smartphones," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2014, pp. 3–14.
18. Z. Liang et al., "SleepExplorer: A visualization tool to make sense of correlations between personal sleep data and contextual factors," *Pers. Ubiquitous Comput.*, vol. 20, no. 6, 2016, pp. 985–1000.

HAMID MANSOOR is currently working toward the Ph.D. degree in computer science with Worcester Polytechnic Institute, Worcester, MA, USA. His research interests include building interactive data visualizations to analyze multivariate data. He is the corresponding author of this article. Contact him at hmansoor@wpi.edu.

WALTER GERYCH is currently working toward the Ph.D. degree in data science with Worcester Polytechnic Institute,

Worcester, MA, USA. His research interests include developing machine learning algorithms to learn from incompletely labeled data. Contact him at wgerych@wpi.edu.

ABDULAZIZ ALAJAJI is currently working toward the Ph.D. degree in data science with Worcester Polytechnic Institute, Worcester, MA, USA. His research interests include developing machine learning algorithms for mobile health. Contact him at asalajaji@wpi.edu.

LUKE BUQUICCHIO is currently working toward the Ph.D. degree in data science with Worcester Polytechnic Institute, Worcester, MA, USA. His research interests include developing machine learning algorithms to understand the emergence of new, unknown classes. Contact him at lbuquicchio@wpi.edu.

KAVIN CHANDRASEKARAN is currently working toward the Ph.D. degree in data science with Worcester Polytechnic Institute, Worcester, MA, USA. His research interests include developing machine learning algorithms for recognizing complex human activities. Contact him at kchandrasekaran@wpi.edu.

EMMANUEL AGU is currently a Professor in the Computer Science Department, Worcester Polytechnic Institute, Worcester, MA, USA. His research interests include computer graphics, mobile computing, image analysis, and machine learning especially applications in healthcare. Contact him at emmanuel@wpi.edu.

ELKE A. RUNDENSTEINER is currently a Professor of computer science and the founding Director of the interdisciplinary Data Science program at Worcester Polytechnic Institute, Worcester, MA, USA. As an internationally recognized expert in big data analytics, her research spans data science, data stream analytics, machine learning, visual and computational big data infrastructures, and digital health. Contact her at rundenst@wpi.edu.

Contact department editor Theresa-Marie Rhyne at theresamarierhyne@gmail.com.



PURPOSE: The IEEE Computer Society is the world's largest association of computing professionals and is the leading provider of technical information in the field.

MEMBERSHIP: Members receive the monthly magazine *Computer*, discounts, and opportunities to serve (all activities are led by volunteer members). Membership is open to all IEEE members, affiliate society members, and others interested in the computer field.

COMPUTER SOCIETY WEBSITE: www.computer.org

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 and 17 journals. Refer to membership application or request information as noted above.

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

Conferences/Education: The society holds about 200 conferences each year and sponsors many educational activities, including computing science accreditation.

Certifications: The society offers three software developer credentials. For more information, visit www.computer.org/certification.

BOARD OF GOVERNORS MEETING

TBD

EXECUTIVE COMMITTEE

President: William D. Gropp

President-Elect: Nita Patel

Past President: Forrest Shull

First VP: Riccardo Mariani; **Second VP:** David S. Ebert

Secretary: Jyotika Athavale; **Treasurer:** Michela Taufer

VP, Membership & Geographic Activities: Andre Oboler

VP, Professional & Educational Activities: Hironori Washizaki

VP, Publications: David S. Ebert

VP, Standards Activities: Annette Reilly

VP, Technical & Conference Activities: Grace Lewis

2021–2022 IEEE Division VIII Director: Christina M. Schober

2022–2023 IEEE Division V Director: Cecilia Metra

2022 IEEE Division VIII Director-Elect: Leila De Florian

BOARD OF GOVERNORS

Term Expiring 2022: Nils Aschenbruck, Ernesto Cuadros-Vargas, David S. Ebert, Grace Lewis, Hironori Washizaki, Stefano Zanero

Term Expiring 2023: Jyotika Athavale, Terry Benzel, Takako Hashimoto, Irene Pazos Viana, Annette Reilly, Deborah Silver

Term Expiring 2024: Saurabh Bagchi, Charles (Chuck) Hansen, Carlos E. Jimenez-Gomez, Daniel S. Katz, Shixia Liu, Cyril Onwubiko

EXECUTIVE STAFF

Executive Director: Melissa A. Russell

Director, Governance & Associate Executive Director: Anne Marie Kelly

Director, Conference Operations: Silvia Ceballos

Director, Information Technology & Services: Sumit Kacker

Director, Marketing & Sales: Michelle Tubb

Director, Membership & Education: Eric Berkowitz

Director, Periodicals & Special Projects: Robin Baldwin

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: +1 800 678 4333; **Fax:** +1 714 821 4641;

Email: help@computer.org

IEEE BOARD OF DIRECTORS

K. J. Ray Liu, *President & CEO*

Saifur Rahman, *President-Elect*

John W. Walz, *Director & Secretary*

Mary Ellen Randall, *Director & Treasurer*

Susan "Kathy" Land, *Past President*

Stephen M. Phillips, *Director & Vice President, Educational Activities*

Lawrence O. Hall, *Director & Vice President, Publication Services and Products*

David A. Koehler, *Director & Vice President, Member and Geographic Activities*

James E. Matthews, *Director & President, Standards Association*

Bruno Meyer, *Director & Vice President, Technical Activities*

Deborah M. Cooper, *Director & President, IEEE-US*

Common Shortcomings in Applying User-Centered Design for Digital Health

Lorraine R. Buis, *University of Michigan*

Jina Huh-Yoo, *Drexel University*

User-centered design (UCD) is the focus of many teams developing digital health innovations, yet principles are not always effectively employed. We have identified several common ways that UCD principles fall short at all stages of the design process and steps to overcome those pitfalls.

From connecting patients, providers, and caregivers, to remotely monitoring patient health outcomes, to aiding individuals self-managing their own conditions, technology is changing how we manage health. As technologies with potential application to healthcare have advanced, so has interest in pushing the envelope.

As experienced researchers of digital health solutions, we have been researching, publishing, and reviewing grants in this space and working with our own multidisciplinary teams of engineers, computer scientists, clinicians, and patients. We have learned many lessons from our successes and failures, and those of others. On one hand, we have seen solutions change lives, but on the other, we have seen innovations make advancements in science and engineering but lack clinical utility. Sometimes these are true “hammers in search of nails,” where an innovation is inadequately applied to healthcare; other times they are misguided attempts at innovation, lacking proper understanding of clinical context.

Through this, we have observed and experienced several common challenges that determine the success of digital health projects. When these challenges arise, these are the projects that do not get funded, do not move forward, or end up sitting on a shelf. When

reflecting back, despite widespread acknowledgment and understanding of user-centered design (UCD) principles in academia and industry, UCD methodologies are often overlooked, given short shrift, or pointedly ignored in the face of different constraints, including lack of time, money, resources, and personnel; they are also ignored because of ego, outright hubris, and the belief that we as innovators know exactly what is needed.

No one is immune from skimping on the UCD process. Despite our own formal training, we also make missteps and regret not following our own advice. Some examples of common situations that lead to oversights include.

- ▶ Funding opportunities with short turnaround times, which may cause teams to guess about the types of solutions users need or want.
- ▶ Engineering labs that make quick progress on solutions with healthy volunteers before moving to feasibility testing in target populations, which often end up with technically innovative solutions that lack clinical utility.
- ▶ Lack of suitable resources (money, time, personnel, etc.), or access to clinicians and patients.
- ▶ Teams seasoned in working with a target population who want to develop new types of solutions without starting from the ground up.

Digital Object Identifier 10.1109/MPRV.2020.2997615

Date of current version 30 July 2020.

The goal of this article is to point out the pitfalls that development teams face when working in the digital health space, structured through the lens of the UCD framework, so that the pitfalls can be avoided in the future and good products can be designed.

COMMON UCD CHALLENGES

Not Getting the Clinical Context Right

Getting the clinical context right is a common challenge experienced during the design of new innovations. Without adequate expertise, we have seen grant proposals focused on all stages of technology development get rejected because of the lack of adequate clinical expertise on the team, or lack of clinical utility. Just because something *can* be developed does not mean it *should* be developed, or that it is the *right* thing to develop. Having suitable clinical expertise on the team can help with the following.

- › Finding the correct clinical domain.
- › Finding the critical symptom, health-behavior, or health-outcome to target.
- › Providing clinical perspective for collecting, interpreting, and monitoring data.
- › Integrating the design with clinical workflow.
- › Understanding feasible design in the clinical context.

Overcome the Pitfall: Assemble the Right Team

To get clinical context right, it is essential to have not just any clinical expertise, but to have the *right* clinical expertise. Building bridges between technical and clinical collaborators can be difficult. Universities with schools or colleges devoted to medicine, nursing, pharmacy, public health, or other allied health professions may have an advantage over universities that do not; however, colocation of technical and clinical departments on a campus does not mean that collaborations naturally emerge. It takes time and care to build trust and understanding, both of which are essential for productive interdisciplinary relationships. It is also not necessarily enough to have clinicians on your team who are experts in their clinical domain. Finding clinicians with experience working with developers, or who are excited to think about, and act on ways to

transform current clinical practice, can be key to having a good, collaborative relationship for building digital health solutions. To determine whether you have the right clinical people on your team, ask yourself the following.

- › Do my clinical team members work with patients who have the target health condition?
- › Are my clinical team members experts in condition-specific pathophysiology and treatment?
- › Do my clinical team members have knowledge of and experience working with the target population?
- › Have my clinical team members conducted research with the condition and population in the past?
- › Have my clinical team members developed digital health solutions before and do they have experience working with developers?

Not Getting the User Context Right

Failure to understand users' context is also commonplace. Even as those who are aware of UCD's fundamental principle—to understand users' needs and contexts, we often overestimate the desire that individuals have to understand and manage their health. We also overestimate the lengths that users are willing to take to improve their health-related behaviors. For example, one common component of digital health solutions is tracking health outcomes. Technology provides opportunities to track data in ways that have never been possible before. We often make assumptions that people *want* to track data for health purposes, because there is an assumption that people *want* to change their behavior or improve their health. This is not always the case. Oftentimes, health-related behaviors are not something people engage in because people *want* to do them; rather they do so because they *have* to.

For instance, food logging is a common component of many digital health solutions; however, getting people to faithfully and accurately record dietary intake over time is hard. Despite evidence that modifying diet can lead to long-term health benefits, our research has taught us that food logging is typically seen by users as burdensome, boring, and unenjoyable. It sounds easy

in theory, but is difficult in practice. Previous research has shown that when asked to log diet, many individuals batch enter and back-fill data instead of tracking in real, or near-real time as intended.¹ This can lead to inaccurate data regarding food, portions, or timing of meals, which may undermine behavior change.

Lack of understanding user beliefs and cultural context also speaks to technology use. Although we may think users will want to provide an endless stream of data that is collected in perpetuity, that is not a realistic ask. Some users may be willing to engage in the short-term, but not the long-term. Even with automated sensor-based tracking, burdens still exist with having to wear sensors and make sure data are flowing appropriately. For some, lack of sustained and faithful tracking occurs because of burden, but for others, the desire to not engage is related to situated social and cultural concerns, such as privacy and confidentiality.

User context can be biased through lack of experience working with target populations, or reliance on students or employees for formative development activities, instead of target end users. The solution we as digital health innovators think is useful for us is not always also useful for users.

Overcome the Pitfall: Get User Context Right

We should get to know the target end users as we start the project, not in hindsight. Leaving the walls of your office and finding your average user on the street takes time and effort, but is worth it to be more user-centered, and will likely result in better design. You can start this process by first asking yourself the following.

- › What am I asking a user to do, and would I (or my family/friends) be willing to do that?
- › How frequently am I asking a user to use my innovation, and would I (or my family/friends) be willing to do that?
- › For how long am I asking a user to use my innovation, and would I (or my family/friends) be willing to use it for that long?
- › How much of a burden am I placing on a user and is that comparable to the benefit they will get?
- › What potential risks am I bringing to users by using my innovation?

Not Properly Identifying User Requirements

Even if we get the context right, we might fail to get the user and their requirements right. Creation of personas that guide development for different groups is a common starting point for design. These personas are archetypes of different types of users, often given names, faces, and a backstory, and are used to delineate tasks and goals that may be accomplished through the use of a product. Although these personas are great for capturing user task context, they may miss essential aspects of users themselves. We think we know the user, but we often do not.

For example, cell phone app development has exploded in parallel with widespread cell phone adoption, even among populations who have historically experienced barriers to adopting technology. However, in our work, we find that users often struggle with the devices they have. They struggle with downloading apps because they do not know their password, technical literacy is questionable, and pairing devices can be an incredible burden to overcome.

In addition to overestimating technical abilities, designers often overestimate users' understanding of their health. We have a lot of information about the information needs of patients who are active in social media; however, this is a biased group of people who enjoy posting about their health online. Not all patients with Type 2 diabetes, for instance, will have as good of a handle on how their diet contributes to blood sugar fluctuations as we expect from seeing what is shared in online diabetes communities.

Furthermore, most use contexts involve multiple stakeholders who influence individuals' health behaviors. The roles of social networks, including caregivers, friends and family, and healthcare teams often go unnoticed, as innovators often focus on users themselves. In reality, the health of an individual is an enormous lesson in situated action as people are influenced by the context they live in, which can have profound effects on health behaviors. For example, if you are trying use an app to improve diet among pregnant adolescents, you must understand that teens often do not have control over the foods they have access to within the home. Likewise, those looking to create a care coordination app for dementia care need to understand that both informal and formal caregivers

(including social workers, physicians, nurses, and nursing home administrators) all influence the health of a patient. As Orlikowsky's Technological Frames work argues,² these multiple stakeholders would have vastly different understandings of what technology adoption might mean for their work context, and unless the new technology solution addresses all of their needs, the adoption may fail.

Overcome the Pitfall: Get User Requirements Right

Although the role of personas is important for formative task-based design work, we must be careful not to reduce target users to archetypes that fail to grasp the essence of what users struggle with and need. Understanding the range of skills and abilities of users is critical. To ensure that we know what the realistic user requirements are, ask the following questions.

- › What ranges of tech savviness do the target users have?
- › Do target users have experiences using the technology platform my solution relies on?
- › Is my target solution suitable for use by my friends/family?
- › Do the target users understand enough about the specific health condition to lead them to understand why behavior change is beneficial?
- › Even if they understand the benefits, do they have any economic, social, or cultural barriers that might hinder change?
- › Have I considered who else may be influencing the behavior of the target users?

Failure to Design the Right Solution

Even if the context is well understood and users and their requirements are well defined, new innovations can fail in the design phase, particularly when it comes to envisioning and refining adequate solutions to a problem. Many projects fail because teams do not understand what the right kind of solution is, and they do not iterate until the solution evolves into something that meets the needs of all stakeholders. Technology vendors in the marketplace constantly develop technically innovative tools for collecting, compiling, and transmitting data from patient-side peripheral remote monitoring devices to providers, with the

goal of revolutionizing care. Although these tools will likely be an essential component in healthcare in the future, our current healthcare system is not ready to use these tools clinically on a broad scale. Not only is our current health IT infrastructure not yet optimized to handle large-scale patient self-monitoring, but our workflow processes are not yet optimized to that model of care.

In our current model of care, physicians are limited in the time they can spend on individual patients. Working with clinicians can help ensure that proposed solutions can fit within the existing model of care. Furthermore, the better we can integrate solutions into current care environments, the more likely we can overcome the often touted 17-year uphill battle of making evidence-based changes to clinical practice.³

Overcome the Pitfall: Get the Design Right

We must take steps to make sure that proposed solutions are amenable and well positioned to be effective for the target population. Ways to ensure that we are on track include asking the following questions.

- › What is the likelihood that my solution will be used by the appropriate people at the designated time?
- › How will users incorporate this into their daily life?
- › What do users need to change about their habits in order to use the solution as intended?

Flaws in Evaluation

Ideally, the team has worked closely with clinical collaborators and user stakeholders from the beginning, and iteratively refined the system according to UCD processes. However, even if good solutions have been designed, we have seen projects make missteps during the evaluation, which can sink a grant application, or prohibit a solution from advancing. This usually occurs when we select the wrong metrics for evaluation. These metrics, which are the criteria used to determine whether the project has succeeded, are often highly dependent on clinical and use contexts. For instance, we have seen evaluations that hinge on irrelevant or suboptimal metrics, or metrics that are so highly confounded that it is impossible to tell if a digital health solution worked.

Overcome the Pitfall: Get the Evaluation Right

We need to make sure that the metrics we use to ascertain success are the measures that will actually tell us if a digital health solution works. Ways to ensure that we are on track include asking the following questions.

- › Do my evaluation metrics tell me critical information about the success of my product?
- › Are my evaluation metrics rooted within the appropriate clinical context?
- › Are my evaluation metrics the best ones for use in this context?

CONCLUSION

The UCD process is just one tool in our toolkit for achieving success; however, it is essential that careful application of the steps be made. We have seen many ways that good development teams have ignored parts of the UCD process, and we too have not been immune from failing to apply the process appropriately. Sometimes, we get lucky and no serious harm befalls the design process, but more often than not, when a project does not pan out, hindsight can show us that failure to adequately engage in UCD was the catalyst. In reflecting on our years of digital health development, our lessons learned can be distilled down pretty easily. Within healthcare, providers, and especially nurses, are familiar with the “5 rights” of medication use, which include the “right patient, right drug, right time, right dose, and right route.” For UCD, it is “users are not like me.” These mantra together emphasize the need for a proper application of UCD principles for digital health development; the right clinical context, the right user context, the right user requirements, the right design, and the right evaluation. 🙌

ACKNOWLEDGMENTS

The authors would like to thank Lilly Pritula and Reema Kadri for help editing this manuscript, as well as Gabriela Marcu, PhD for guidance and support.

REFERENCES

1. L. E. Burke, et al., “Using instrumented paper diaries to document self-monitoring patterns in weight loss,” *Contemp. Clin. Trials*, vol. 29, no. 2, pp. 182–193. Mar. 2008.
2. J. W. Orlikowski and C. D. Gash, “Technological frames: Making sense of information technology in organizations,” *ACM Trans. Inf. Syst.*, vol. 12, no. 2, pp. 174–207. Apr. 1994.
3. Z. S. Morris, S. Wooding, and J. Grant, “The answer is 17 years, what is the question: Understanding time lags in translational research,” *J. R. Soc. Med.*, vol. 104, no. 12, pp. 510–520. Dec. 2011.

LORRAINE R. BUIS is an assistant professor with the Department of Family Medicine and the School of Information, University of Michigan, Ann Arbor, MI, USA. Contact her at buisl@umich.edu.

JINA HUH-YOO is an assistant professor of Human-Computer Interaction with the Department of Information Science in the College of Computing and Informatics, Drexel University, Philadelphia, PA, USA. Contact her at jh3767@drexel.edu.



Drive Diversity & Inclusion in Computing



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



Do you have a great idea for new programs that will positively impact diversity, equity, and inclusion throughout the computing community?

The IEEE Computer Society Diversity & Inclusion Committee seeks proposals for projects, programs, and events that further its mission. New programs that deliver education, outreach, and support, including, but not limited to, mentoring programs at conferences, panel discussions, and webinars, are welcomed.

Help propel the Computer Society's D&I programs—submit a proposal today!

<https://bit.ly/CS-Diversity-CFP>



Donations to the IEEE Computer Society D&I Fund are welcome!



**IEEE
COMPUTER
SOCIETY**

IEEE Foundation

Security and Privacy for Edge Artificial Intelligence

James Bret Michael, *Associate Editor in Chief*

Edge artificial intelligence (AI) takes decentralization of data and computing to a new level, providing for optimization of resource allocation, and development and functioning of AI, on edge devices. It also introduces opportunities and challenges in the realm of security and privacy.

The convergence of AI, edge computing, and cloud computing impacts our daily lives. For instance, the smart thermostat I recently installed in my home has been learning what my family considers to be a comfortable temperature for different times of the day and night. The thermostat is an edge device that takes as input the sensed room temperature and temperature settings manually input by my family members, refines the machine learning (ML) algorithm on this data, and then makes inferences using the algorithm to select the output control signals to send to my home's furnace. The automation of the temperature-adjusting function with in-situ learning has already resulted in energy savings and the house being neither too warm nor too cool.

Devices like the smart thermostat are part of what is known as *edge AI* or *edge intelligence*. In this "From the Editors" column, I use the former term. But let's back up a moment. What is edge computing? It is a form of distributed computing in which an edge device, or a set of neighboring edge devices, performs computing tasks that would otherwise be done on remote cloud servers. Why is edge computing appealing? With the rapid growth of the Internet of Things (IoT), there is now a vast amount of data being sensed and produced at the edge so enormous in size that it is

not technically feasible using the bandwidth of today's Internet to transfer the entirety of the data from the edge devices to cloud servers for storage and processing; even if the bandwidth was available, there would need to be enough data center resources available to handle all of the data. In addition to bandwidth issues, the communication latency incurred by treating edge devices as clients of cloud servers can make it impractical to meet user requirements for fast reaction or response times, such as supporting the real-time decision making performed by collision-avoidance systems embedded in passenger vehicles, buses, and trucks. In other words, there is a need for some degree of federated intelligence. The edge devices, the cars in this example, need local processing for much (but not all of) their activity. Some tasks performed by edge devices may require a combination of local and remote processing.

EDGE AI COMPARED TO EDGE COMPUTING

What differentiates edge AI from edge computing? Edge AI incorporates AI capabilities on the edge devices. Deng et al. partition edge AI into two categories: AI for edge (also known as *intelligence-enabled edge computing*) and AI on edge.¹ The former is concerned with optimizing the allocation of resources used at the edge, whereas the latter includes "carry[ing] out the entire process of building [and running] AI models on the edge." The smart thermostat serves as an example of AI on edge: this IoT device updates the ML algorithm, makes inferences, and decides what actions to take to shape the behavior of the heating, ventilation, and air-conditioning system. One of the benefits of edge AI technology is that the data needed for refining the algorithm is decentralized. Another advantage

Digital Object Identifier 10.1109/MSEC.2021.3078304
Date of current version: 1 July 2021

is that analysis and decision making can be performed close to the source of the data. From a security and privacy perspective, edge AI can remove attack vectors by minimizing or eliminating the transfer of data between the edge devices and their data centers. In the case of my home thermostat, however, there is likely a continuing connection between my thermostat and some vendor-operated server. Thus, it is likely that my smartphone app is communicating with a central server, which is collecting data and using a session that was previously set up by the thermostat when it connects to the server. I did not set up port forwarding on my router (this has its own security issues).

There are security and privacy issues that arise with the use of edge AI. My thermostat transmits information about its reliability, its performance with optimizing the use of the furnace under multiple constraints, and the parameters of its ML model to cloud servers for use by the company that manufactured the device, ostensibly for improving the company's product line of smart thermostats. How much pattern-of-life information about my family can be deduced from these data? Who has access to them and why? I do not know whether any raw data (for example, unprocessed sensor readings), personally identifiable information about my family and me, or details about my home network security settings (e.g., router password and firewall settings) are shared with the company. I also have no knowledge of how the company transfers those data. Does it use secure connections? Does the company protect this information from side-channel attacks? Finally, does the company update the ML model's parameters from afar (that is, by pushing those parameters from on-cloud servers) without notifying me?

I can remotely communicate with the thermostat from my smartphone via my home's wireless network and the Internet, such as to check on the temperature of my home and change temperature settings, but how much trust should, or can, I place in the authentication and other protocols used with the remote-access functions of the smartphone app for the thermostat? Another issue is that I do not know how much trust to place in my home's wireless infrastructure. I recently received a message from the manufacturer of the routers that I use stating that I need to perform firmware updates. What actually happened when I tried

to apply the updates was that the company installed an unwanted network security-scanning application. I lump this into the category of a supply-chain risk. But let's get back to the topic at hand.

In addition, the smart thermostat can operate autonomously. As I mentioned, it learns to self-regulate the temperature inside the home, with no need for attention or input by the user. After about the third week of the thermostat's operation, I stopped fiddling with and checking on the device. I had no situational awareness of what the thermostat was doing, other than that the temperature in the home was comfortable. I stopped logging into the app. I also did not bother to change my password after the initial installation. However, I did set up two-factor authentication. But what if someone hacks my smart

IEEE SECURITY & PRIVACY WELCOMES
SUBMISSION OF ARTICLES ON THIS
FASCINATING, RAPIDLY ADVANCING,
AND GAME-CHANGING TECHNOLOGY.

thermostat app on my phone and disables the thermostat or hacks the device itself? If someone hacks into my thermostat, then my network is vulnerable, and possibly my family's computing devices become targets—likely of more interest than the thermostat. Could the attacker start a fire or other significant disruption in the house? (Turning the furnace on and off repeatedly over a short period of time might cause it to fail in unexpected ways.)

Before I purchased the thermostat, I was aware that it and other IoT devices are not immune to attack: they are tempting targets for cybermischief, and hackers have demonstrated the exploitation of vulnerabilities in edge devices designed for the home.² I accepted the security and privacy risks for the convenience of automating mundane home-management tasks. I also installed an IoT-based digital door lock. The lock has no AI functionality, so it is an example of edge computing but not edge AI. That is the extent, so far, of my foray into home automation.

Furthermore, with my thermostat falling into the category of AI on edge, there is the possibility that

someone may launch an ML adversarial attack, poisoning the data used by the ML algorithm. However, someone could also perform a physical adversarial attack by just leaving a window open near the thermostat. The result of such an attack would likely be the thermostat behaving in an unexpected way, such as causing wide swings in temperature or short-cycling the furnace (that is, causing it to rapidly turn on and off), ultimately resulting in damage to the furnace and an inefficient use of energy. I am not overly concerned about the risks to security and privacy posed by my home-based IoT devices, but I am concerned about those types of issues for industrial and national security uses of edge AI, for which the stakes are higher in the event of compromised security or privacy.

Let's consider a use case for the Industrial IoT. Chemical manufacturing plants, such as those used in the refining of oil and gas, are instrumented with lots of sensors and microcontrollers, with the aim of collecting trusted data to be analyzed as a part of optimizing the processing of the chemicals, maintaining product quality, and monitoring the safety and security of the plant operations. Decentralizing the data provided by the sensors and microcontrollers, in addition to applying AI on edge, is already happening in the chemical manufacturing industry and is viewed as being a vital means for companies to gain an edge—pardon the pun—over their competitors.³

For this use case, consider the following scenario. A manufacturing plant produces hydrogen sulfide (H_2S), a colorless chalcogen hydride gas also known as *hydrosulfuric acid*. H_2S is poisonous, corrosive, and flammable. Further, let's suppose that the plant employs a federated approach to ML at the edge, with AI-on-edge devices located near sensors that monitor the storage-and-feed, reactor, and clean-up sections of the production unit. In federated ML, each device uses its own "local data to cooperatively train an ML model required by a federated learning (FL) server. They then send the model updates, i.e., the model's weights to the FL server for aggregation. The steps are repeated in multiple rounds until a desirable accuracy is achieved."⁴

What can possibly go wrong? Well, for one thing, it is conceivable that the server acting as the aggregator could leak the trained model or information about the local data sets. In addition, as pointed out by Lim et al.,

malicious participants (that is, one or more compromised edge devices) could poison the data and model by, for example, "send[ing] incorrect parameters or corrupted models to falsify the learning process during global aggregation."⁴ The hacker's intent might be to cause the inference models to improperly manage the cooling water exchangers needed to keep the reactor or clean-up units from overheating, potentially resulting in a mishap if the plant fails to contain the H_2S product. There is the age-old problem of deciding which of the devices can be trusted.

The foregoing example could have been made more complex, allowing for the distributed system to be composed of heterogeneous edge devices (this includes sensors) and cloud servers, along with data sources of varying and possibly unknown levels of quality and trust. What do such systems portend for the specification and implementation of policy and mechanisms for security and privacy?

Edge AI is a burgeoning area of research and development, in part because the enabling technologies are becoming available, such as 5G networks, high-performance AI chips,⁵ lightweight AI models, AI-specific service architectures for the edge,⁶ co-design methodologies tailored for edge computing and edge AI,⁷ and lightweight and leakage-resilient authenticated key exchange protocols for edge AI.⁸ Lim et al. are investigating ways to apply edge AI to large-scale mobile edge networks of heterogeneous devices while preserving the privacy of the data of each of the participants (edge nodes) that are taking part in FL in the presence of one or more malicious participants or aggregators (servers).⁴ They have explored several solutions to the problem of not being able to assume that all participants and aggregators can be trusted, such as schemes for secure aggregation and differential privacy. As another example, Libri et al. have demonstrated the application of edge AI by using large-scale sensor networks to detect malware in data centers.⁹

On a personal note, in the mid-1990s, I was a research engineer with the University of California, Berkeley's California Partners for Advanced Transit and Highways program. My colleagues and I were at the forefront of mobile edge computing, demonstrating in 1997 the technical feasibility of safely operating

dual-mode automobiles under fully automated control in platoon formations under high-performance driving conditions (e.g., maintaining a velocity of 30 m/s with as little as one car length between vehicles) on dedicated highway lanes.¹⁰ We used classical control system technology to implement the system, with a four-layer hierarchical control architecture: network, link, planning, and regulation.¹¹ The individual vehicles processed their own sensor data and the data shared among the vehicles. The vehicles also communicated with the instrumented roadway infrastructure. AI on edge would have been helpful, used in concert with the design of the controllers, for developing and continuously improving the models and algorithms used to achieve optimizations, such as for lane-change maneuvers, emergency braking and other safety-related actions under various environmental conditions, minimizing sulfur and nitrogen oxides emissions, and reaching levels of throughput of vehicles that come close to the theoretical capacity of dedicated lanes on the automated highway system.

As edge AI advances and significant progress is made in the evolution from weak to strong AI (that is, from custom AI systems that are tailored to a specific application or limited number of tasks to AI system that have general intelligence abilities), it will be interesting to see how existing security and privacy risks are handled and what new risks and opportunities arise. *IEEE Security & Privacy* welcomes submission of articles on this fascinating, rapidly advancing, and game-changing technology. Please also keep your eyes open for an upcoming call for papers for a theme issue of the magazine on this subject. 🤖

DISCLAIMER

The views and conclusions contained herein are those of the author's and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. government.

REFERENCES

1. S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya, "Edge intelligence: The confluence of edge computing and artificial intelligence," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7457–7469, 2020. doi: 10.1109/JIOT.2020.2984887.
2. R. Albergotti, "How Nest, designed to keep intruders out of people's homes, effectively allowed hackers to get in." *Washington Post*, Apr. 23, 2019. <https://www.washingtonpost.com/technology/2019/04/23/how-nest-designed-keep-intruders-out-peoples-homes-effectively-allowed-hackers-get/> (accessed May 3, 2021).
3. S. Ottewell, "IIoT: Chemical makers approach the edge." *Chemical Processing*, Mar. 10, 2020. <https://www.chemicalprocessing.com/articles/2020/iiot-chemical-makers-approach-the-edge/> (accessed May 3, 2021).
4. W. Y. B. Lim et al., "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031–2063, 2020. doi: 10.1109/COMST.2020.2986024.
5. A. James, "The why, what and how of artificial general intelligence chip development," *IEEE Trans. Cogn. Devel. Syst.*, early access, 2021. doi: 10.1109/TCDS.2021.3069871.
6. S. Kum, Y. Kim, D. Siracusa, and J. Moon, "Artificial intelligence service architecture for edge device," in *Proc. 10th Int. Conf. Consumer Electron.*, 2020, pp. 1–3. doi: 10.1109/ICCE-Berlin50680.2020.9352184.
7. C. Hao, J. Dotzel, J. Xiong, L. Benini, Z. Zhang, and D. Chen, "Enabling design methodologies and future trends for edge AI: Specialization and co-design," *IEEE Des. Test.*, early access, 2021. doi: 10.1109/MDAT.2021.3069952.
8. J. Zhang, F. Zhang, X. Huang, and X. Liu, "Leakage-resilient authenticated key exchange for edge artificial intelligence," *IEEE Trans. Dependable Secure Comput.*, early access, 2020. doi: 10.1109/TDSC.2020.2967703.
9. A. Libri, A. Bartolini, and L. Enini, "pAElla: Edge AI-based real-time malware detection in data centers," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9589–9599, 2020. doi: 10.1109/JIOT.2020.2986702.
10. "National automated highway system consortium technical feasibility demonstration summary report," National Automated Highway System Consortium, Troy, MI, Feb. 1998. Accessed: May 4, 2021. [Online]. Available: https://path.berkeley.edu/sites/default/files/part_1_ahs-demo-97.pdf
11. P. Varaiya, "Smart cars on smart roads: Problems of control," *IEEE Trans. Autom. Control*, vol. 38, no. 2, pp. 195–207, 1993. doi: 10.1109/9.250509.

Edge Artificial Intelligence Chips for the Cyberphysical Systems Era

Hiroshi Fuketa and Kunio Uchiyama, *National Institute of Advanced Industrial Science and Technology, Japan*

Artificial intelligence (AI) chips draw much attention for cyberphysical systems since AI chips are promising to realize edge AI computing. We introduce the chip architecture that enables energy-efficient computing and design tools for AI chips.

The dramatic progress of artificial intelligence (AI) in recent years is mainly the result of advances in deep learning (DL) algorithms. DL-based AI is used now in a wide variety of applications, such as image recognition and machine translation. However, it is reported that the amount of computation required to run these algorithms has been exponentially increasing, with a 3.4-month doubling time since 2012.¹ Therefore, AI chips, which can compute DL algorithms more efficiently compared to conventional CPUs and GPUs, have been drawing much attention. [In this column, we use “AI chips” as a generic term for DL accelerators, and hence the following two types of implementations for AI chips are considered: 1) independent large-scale integrations, including Google’s tensor processing unit, and 2) intellectual property (IP) cores, such as the neural engine in Apple’s A series system on chip.]

For example, Google and Amazon have been developing original AI chips for their own data centers. These chips are called *cloud AI chips*, and they are used for training deep neural network (DNN) models as well as for inference using DNN models when the amount of computation is too large for edge devices. However, for cyberphysical system (CPS) applications, such as autonomous driving and factory automation (Figure 1), it is critical to conduct AI processing on edge

devices since performing the work on cloud servers entails overhead related to the communication time between the edge and the cloud and the power that is consumed, which is often a crucial factor in CPSs. Therefore, “edge AI chips” that enable AI processing on edge devices are demanded for CPSs. For example, Tesla recently developed an original edge AI chip for autonomous driving. One of the most important requirements for edge AI chips is energy efficiency since edge device power budgets are usually severe. Therefore, various processor architectures to perform DL algorithms on edge devices with high energy efficiency have been recently proposed.

In this column, we describe the architecture of edge AI chips, and we introduce tools that help engineers design such edge AI chips. We mainly focus on image recognition tasks, which are required in typical CPS applications such as autonomous driving and factory automation.

AI CHIP ARCHITECTURE FOR ACHIEVING HIGH ENERGY EFFICIENCY

Architecture suitable for computation in neural networks

A DNN consists of many stacked layers of neural networks. As a typical neural network architecture, Figure 2 shows a fully connected (FC) neural network and a convolutional neural network (CNN). In the FC version, the output activations are calculated by the

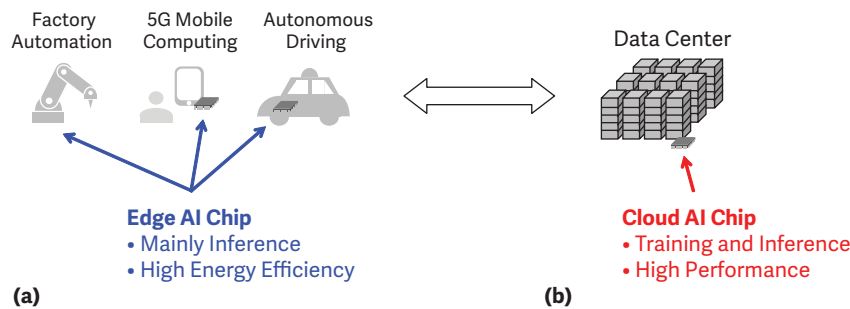


FIGURE 1. AI chips for (a) the edge and (b) the cloud in CPSs.

weighted sum of the input activations [Figure 2(a)]. As the number of input and output activations increases, a large amount of memory capacity and bandwidth for weights and computation is required. On the other hand, a CNN is often used for image recognition tasks. CNNs consist of multiple filters, and the output feature map is calculated by sliding those filters in the input feature map, as represented in Figure 2(b).

The FC and CNN calculation methods are described at the bottom of Figure 2(a) and (b). As indicated by these equations, the calculations mainly consist of multiply and accumulate (MAC) operations. It is well known that the performance of MAC operations can be improved by parallelization. In common GPUs, massive arithmetic logic units (ALUs) are implemented based on the temporal architecture,² as in Figure 2(c), and operate in parallel, which makes it possible to perform MAC operations fast. Therefore, GPUs are widely used for DNN processing.

Another form of parallelization is the spatial architecture (dataflow processing),² as depicted in Figure 2(d). In this architecture, processing elements (PEs) are organized in tiles, and each one consists of an ALU, a register file (RF), and a control circuit. In the temporal architecture [Figure 2(c)], the ALU reads the input data from and writes the calculation result back to the shared RF, whereas in the spatial architecture [Figure 2(d)], data (activations, weights, and partial

sums) can be moved from one PE to another, which reduces the memory access energy requirement. For example, the values of filters (weights) are used many times for computation in a CNN, as in Figure 2(b). The memory access can be reduced by 1) storing these values to the RF in a PE and reusing them and 2) transferring partial sums from one PE to another. This makes computation more energy efficient, and hence the spatial architecture is suitable for edge AI chips.

Reducing computational precision

The most effective way to improve energy efficiency is to reduce computational precision. For common computational tasks, 32-bit floating point (FP32) precision is usually used. In contrast, it is well known that the degradation of computational accuracy is ignorable for DNN tasks (training and inference), even if a lower precision than FP32 is used. For training, 16-bit FP (FP16) precision is often used, and hence many GPUs currently support FP16. In addition, recent studies show the possibility of reducing the computational precision to 8–9 bits for multiplication.^{3,4}

On the other hand, a further reduction of the computational precision is feasible for the inference. For example, Guo et al.⁵ reported that the image recognition accuracy deteriorates by less than 3%, even if the computational precision is altered to an 8-bit integer

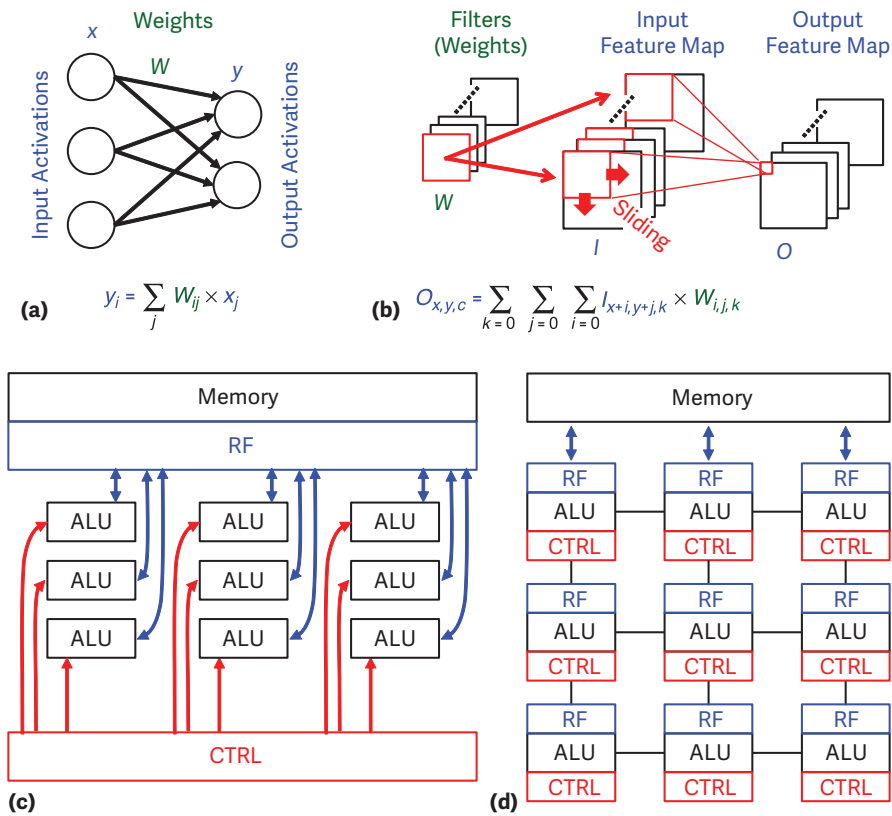


FIGURE 2. (a) and (b) The network architecture and (c) and (d) a form of parallelization to accelerate the computation of neural networks² (a) FC, (b) CNN, (c) the temporal architecture (GPU), and (d) the spatial architecture (AI chip). RF: register file; ALU: arithmetic logic unit; CTRL: control circuit.

from FP32. At this time, the computation energy is significantly reduced; the energy of the 8-bit integer adder and multiplier is 1/30 and 1/19 times smaller than that of the FP32 adder and multiplier, respectively.⁶ This means that reducing the computational precision is very attractive to attain high energy efficiency. Therefore, many edge AI chips that support the low-precision format of a 4/8/16-bit integer have recently been developed since edge AI computing mainly focuses on the inference, as detailed in Figure 1. Figure 3(a) presents the power dissipation of edge AI chips as a function of the performance [in tera operations per second (TOPS)], with their energy efficiency (TOPS/W). By reducing the computational precision to a 4/8-bit integer, a high energy efficiency of several TOPS/W can be achieved.

Recently, many researchers have explored a more aggressive reduction of the computational precision to 1 bit (binary), which is the lowest that is possible.

Neural networks with a 1-bit precision are called *binarized neural networks (BNNs)*.⁷ In BNNs, a single XNOR gate is substituted for a multiplier, and a population counter that tallies the number of ones in an input word is used as an accumulator [as shown in Figure 3(b)], which makes the implementation of the MAC unit much simpler. Thus, extremely high energy efficiency can be achieved by a BNN. For example, Intel developed an AI chip dedicated to BNNs and demonstrated that a significantly high energy efficiency of 617 TOPS/W can be achieved.⁸ However, one of the disadvantages of BNNs is that the networks' inference accuracy degrades due to the reduction of the computational precision, especially for complicated tasks. For instance, the inference accuracy when using BNNs deteriorates by only 1% compared to FP32 precision for the "CIFAR10" small photo classification task, whereas the accuracy worsens by 16% for the more complicated "ImageNet" image classification

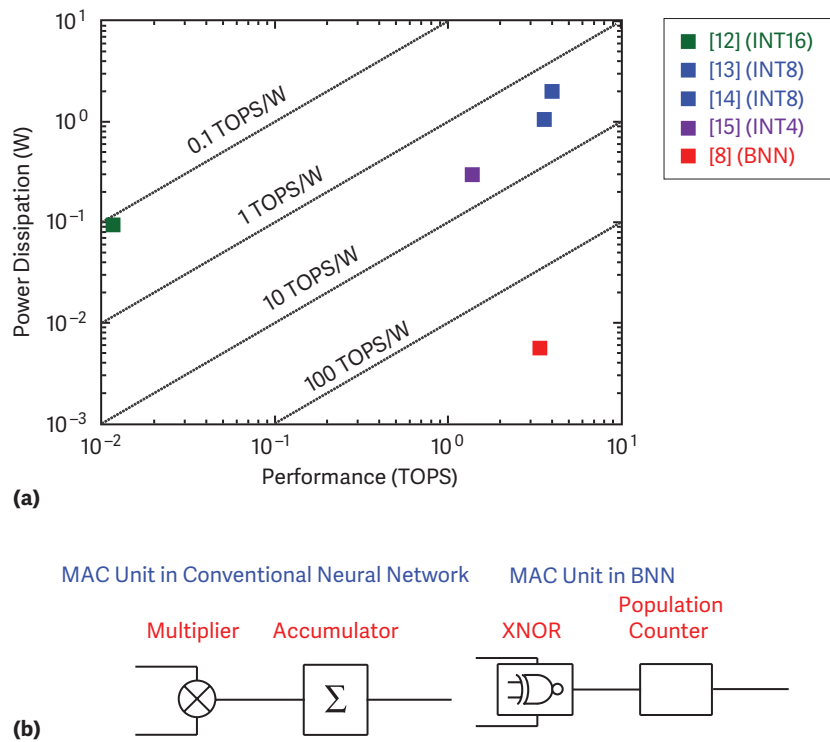


FIGURE 3. (a) The energy efficiency of edge AI chips. (b) The implementation of a MAC unit. INT: integer; BNN: binarized neural network.

task.^{7,9} A practical solution for this issue is to optimize the computational precision at each layer of a neural network. This makes it possible to achieve both high accuracy and energy efficiency.

DESIGN TOOLS FOR EDGE AI CHIPS

Almost all engineers develop DL algorithms by using Python. On the other hand, AI chips are usually devised through a hardware description language, such as Verilog. It is desirable to more easily or automatically create hardware optimized for DL algorithms written in Python. Thus, various design tools for AI chips have recently been developed. As examples, we will describe two tools that focus on developing field-programmable gate array (FPGA)-based edge AI chips.

1. *Vitis AI*: Xilinx has been developing the Vitis AI development environment for its own FPGA products. Vitis AI supports major DNN frameworks written in Python, such as TensorFlow and PyTorch, and offers various tools that

optimize DNN models for hardware implementations, such as 1) pruning, which reduces the model parameters (the number of weights), with a minimal impact on accuracy, and 2) quantization, which lessens the computational precision, from FP32 to the 8-bit integer.

2. *GUINNESS*: This is a graphical user interface-based framework that provides bitstream generation for a Xilinx FPGA. One of the advantages of GUINNESS10 is that it supports BNNs (the details were explained in the previous section), which enables more energy-efficient computing.

Please note that these tools target FPGAs only. One of the reasons is that, currently, the research and development of DL algorithms are mainly conducted by software engineers. FPGAs are appropriate devices for those engineers to use to try to accelerate DL algorithms on custom hardware since the cost of the FPGA development environment, i.e., design tools and the evaluation board, is reasonable.

Although FPGAs are easy to use, their performance and energy efficiency are less than those of application-specified integrated circuits (ASICs). Thus, ASICs for AI are preferable under performance- and power-constrained situations. In particular, ASICs offer the most efficient solution in application domains such as edge computing in CPSs, on which we mainly focused in this column, in terms of power, performance, and cost. However, expensive design environments, such as electronic design automation tools, IPs, and emulators, are required to design AI ASICs, which is a big hurdle for start-ups and small enterprises. To overcome this hurdle, for example, the AI chip design center¹¹ funded by the Japanese government provides a design and verification environment for accelerating the development of AI chips in that country. Through such measures, it is expected that various edge AI chips will be developed in the near future, which will promote the growth of CPSs. 🌟

REFERENCES

3. "AI and compute." OpenAI. <https://openai.com/blog/ai-and-compute/> (accessed Nov. 19, 2020).
4. V. Sze, Y. Chen, T. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proc. IEEE*, vol. 105, no. 12, pp. 2297–2329, 2017. doi: 10.1109/JPROC.2017.2761740.
5. N. Wang, J. Choi, D. Brand, C.-Y. Chen, and K. Gopalakrishnan, "Training deep neural networks with 8-bit floating point numbers," in *Proc. NeurIPS*, 2018, pp. 7675–7684.
6. S. O'uchi et al., "Image-classifier deep convolutional neural network training by 9-bit dedicated hardware to realize validation accuracy and energy efficiency superior to the half precision floating point format," in *Proc. 2018 IEEE Int. Symp. Circuits Syst. (ISCAS)*, pp. 1–5. doi: 10.1109/ISCAS.2018.8350953.
7. K. Guo et al., "From model to FPGA: Software-hardware co-design for efficient neural network acceleration," in *Proc. 2016 IEEE Hot Chips 28 Symp. (HCS)*, pp. 1–27. doi: 10.1109/HOTCHIPS.2016.7936208.
8. S. Han, "Efficient methods and hardware for deep learning," Stanford University, Stanford, CA, 2017. [Online]. Available: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture15.pdf
9. M. Courbariaux et al., "Binarized neural networks: Training neural networks with weights and activations constrained to +1 or -1," 2016. [Online]. Available: arXiv:1602.02830
10. P. C. Knag et al., "A 617TOPS/W All Digital Binary Neural Network Accelerator in 10nm FinFET CMOS," in *Proc. 2020 IEEE Symp. VLSI Circuits*, pp. 1–2. doi: 10.1109/VLSICircuits18222.2020.9162949.
11. M. Rastegari et al., "XNOR-Net: ImageNet classification using binary convolutional neural networks," 2016. [Online]. Available: arXiv:1603.05279
12. H. Nakahara, H. Yonekawa, T. Fujii, M. Shimoda, and S. Sato, "GUINNESS: A GUI based binarized deep neural network framework for software programmers," *IEICE Trans. Inf. Syst.*, vol. E102.D, no. 5, pp. 1003–1011, 2019. doi: 10.1587/transinf.2018RCP0002.
13. AI Chip Design Center. (in Japanese). <https://www.ai-chip-design-center.org> (accessed Nov. 19, 2020).
14. Y.-H. Chen, T. Krishna, J. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," in *Proc. IEEE Int. Conf. Solid-State Circuits (ISSCC)*, pp. 262–264, Feb. 2016. doi: 10.1109/ISSCC.2016.7418007.
15. Google, *Edge TPU*, 2018. <https://cloud.google.com/edge-tpu> (accessed Nov. 19, 2020).
16. C. Lin et al., "7.1 A 3.4-to-13.3TOPS/W 3.6TOPS dual-core deep-learning accelerator for versatile AI applications in 7nm 5G smartphone SoC," in *Proc. 2020 IEEE Int. Solid-State Circuits Conf. (ISSCC)*, San Francisco, CA, 2020, pp. 134–136. doi: 10.1109/ISSCC19947.2020.9063111.
17. J. Lee, C. Kim, S. Kang, D. Shin, S. Kim, and H. Yoo, "UNPU: A 50.6TOPS/W unified deep neural network accelerator with 1b-to-16b fully-variable weight bit-precision," in *Proc. 2018 IEEE Int. Solid-State Circuits Conf. (ISSCC)*, San Francisco, CA, 2018, pp. 218–220. doi: 10.1109/ISSCC.2018.8310262.

HIROSHI FUKETA is a senior researcher at the Device Technology Research Institute and AI Chip Design Open Innovation Laboratory, National Institute of Advanced Industrial Science and Technology, Tokyo, Japan. He is a Member of IEEE. Contact him at h-fuketa@aist.go.jp.

KUNIO UCHIYAMA is an invited senior researcher at the AI Chip Design Open Innovation Laboratory, National Institute of Advanced Industrial Science and Technology, Tokyo, Japan. He is a Fellow of IEEE. Contact him at kunio.uchiyama@aist.go.jp.

IEEE Computer Society Has You Covered!

WORLD-CLASS CONFERENCES — Stay ahead of the curve by attending one of our 210 globally recognized conferences.

DIGITAL LIBRARY — Easily access over 800k articles covering world-class peer-reviewed content in the IEEE Computer Society Digital Library.

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

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

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

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

Explore all of the member benefits at www.computer.org today!



Cognitive Digital Twins for Smart Manufacturing

Muhammad Intizar Ali , Dublin City University Glasnevin, Dublin 9, D09 HXT3, Ireland

Pankesh Patel, University of South Carolina, Columbia, SC, 29208, USA

John G. Breslin , NUI Galway, Galway, H91 TK33, Ireland

Ramy Harik and Amit Sheth , University of South Carolina, Columbia, SC, 29208, USA

Smart manufacturing or Industry 4.0, a trend initiated a decade ago, aims to revolutionize traditional manufacturing using technology-driven approaches. Modern digital technologies such as the Industrial Internet of Things (IIoT), Big Data analytics, augmented/virtual reality, and artificial intelligence (AI) are the key enablers of new smart manufacturing approaches.

The digital twin is an emerging concept whereby a digital replica can be built of any physical object. Digital twins are becoming mainstream; many organizations have started to rely on digital twins to monitor, analyze, and simulate physical assets and processes.¹ The current use of digital twins for smart manufacturing is largely limited to i) *status monitoring*, ii) *simulation*, and iii) *visualization*. For status monitoring, digital replicas of physical assets (e.g., machines) are created, machines are continuously monitored using IIoTs, and the latest status of a machine can be assessed by querying its digital twin. For simulation, digital twins of machines, processes, and products are created to mimic real settings. Simulation allows the design, development, and testing of new products and processes using their digital twins before applying them to actual physical assets, this is presented in.⁵ For visualization, digital twins can include real-time dashboards and alert systems to monitor and debug an operational environment.² However, in contemporary cases, digital twins are simply considered to be an exact replica of the physical assets, without any value-added services built on top of them which could

convert physical assets into autonomous intelligent agents. A major advantage of this enhanced design of digital twins is that they can offer much more than just an exact replica to support value-added services on top of digital twins, which are not possible on the physical assets.

COGNITIVE DIGITAL TWINS

Cognitive digital twins are an extension of existing digital twins with additional capabilities of communication, analytics, and intelligence in three layers: i) *access*, ii) *analytics*, and iii) *cognition*.

The *access layer* is responsible for communication with the machine and gets access to data regarding the status of a physical asset to update the status of the digital twin. The *analytics layer* provides edge analytics capabilities at the device level. Similar to the edge analytics at the edge, this layer of the digital twin can perform additional analytical tasks on top of real-time collected data to help with the process of decision making by converting the raw sensory input into actionable knowledge.³ The *cognitive layer* enables cognition by the digital twins. It is capable of performing complex decision making using edge analytics, domain expertise, and global knowledge bases. It is also responsible for communication among digital twins, allowing them to build their own networks and perform autonomous decision making. Cognitive digital twins will convert traditional digital twins into smart and intelligent agents that can access, analyze, understand, and react to their current status. In case of anomalies, rather than resorting to a simple alert system, the cognitive digital twin can interact with the operational environment and digital twins of products, running processes to further analyze and intelligently understand the anomalies. The cognitive digital twin can draw conclusions of situations locally and then

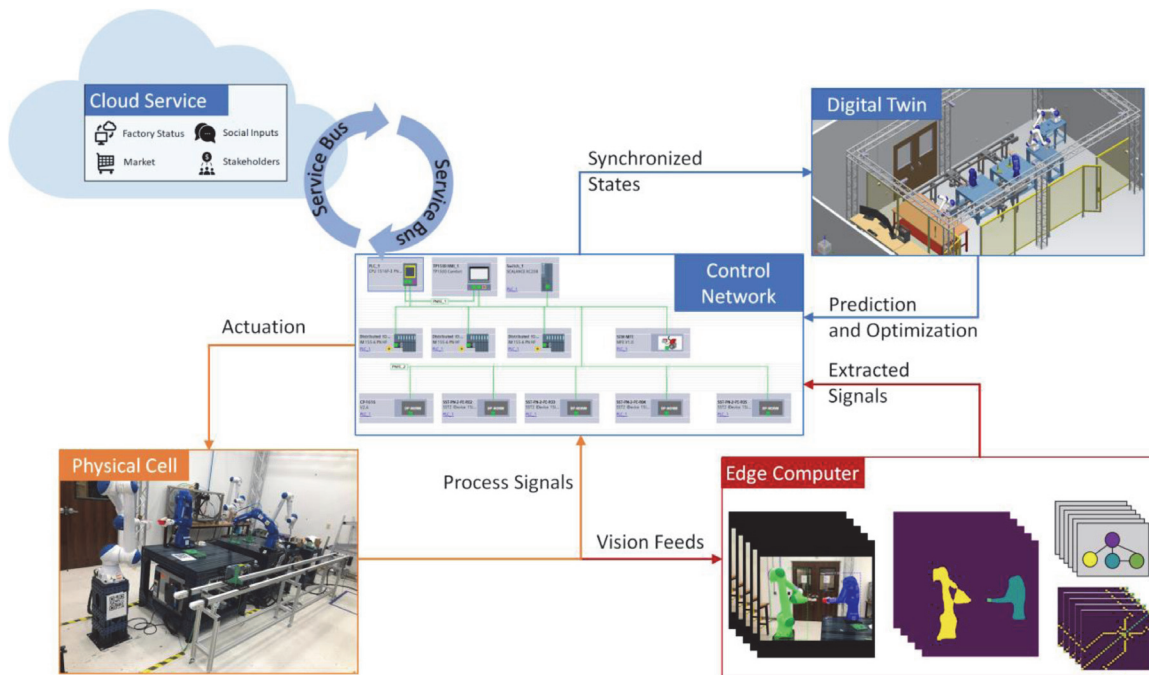


FIGURE 1. Proposed CPS-enabled control for future factories: control network administers physical cell and digital twin to synchronize process signals and intelligently actuate field devices by system smart layers. System smart layers consist of business intelligence from cloud services and semantic integration of visual signals from the edge ends.

also interact with other digital twins of physical assets operating in similar operational conditions to better understand shared local anomalies. Once identified, cognitive digital twins can interact socially with other peers and share knowledge and generate alerts in advance of any future potential unexpected situations. Insights from the analytics performed by cognitive digital twins will eventually help to build enterprise-level knowledge graph extraction, capture, and storage of domain knowledge.

Cognitive digital twins will disrupt existing technologies and applications used for digital twins by making them intelligent as well as social. The emerging concept of self-healing, self-configuring, and self-orchestrating systems is made possible using this approach. The team at the Confirm SFI Research Centre for Smart Manufacturing has implemented an initial prototype of cognitive digital twins using a benchmark dataset for production line performance monitoring⁶ and intend to fully test the implemented prototype on the actual production lines of a smart factory in collaboration with an industry partner. An initial factory of the future to assess and implement this emerging concept is also being constructed at the University of

South Carolina (Figure 1, see Xia *et al.*⁷ for details). Having a social and interactive network of digital twins and a shared knowledge space will allow analytics and intelligence to go beyond the physical walls of a factory where digital twins can share their experience and lessons learned across the board.

ECOSYSTEM OF COGNITIVE DIGITAL TWINS

We envision that once the cognitive digital twins are in place, they can build a network among themselves, having fully automated machine-to-machine interaction and decision making resulting in an ecosystem of cognitive digital twins. The knowledge gained by edge analytics, communication among digital twins, and domain knowledge including user experiences will be captured as a unified knowledge graph. This knowledge graph will gradually evolve and will become a major source of information within the ecosystem of cognitive digital twins. Figure 2 presents a generic overview of cognitive digital twins ecosystems. We further elaborate our vision with an example use case of a manufacturing plant producing orthopedic implants,

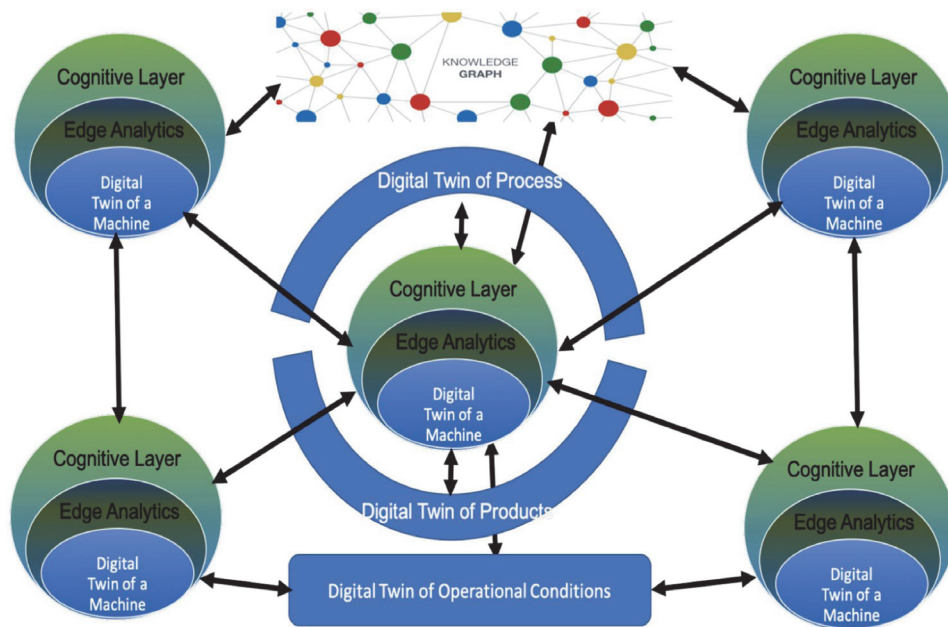


FIGURE 2. Cognitive ecosystem of digital twins.

e.g., knee, hip, and elbow joint replacements. On the shop floor, various machines are placed in an assembly line performing different operations, e.g., cutting, grinding, and polishing, etc. Each machine is equipped with different sensors to monitor its functional state, e.g., temperature, voltage, vibration, and rotation. A cognitive digital twin is created for all machines, products, and processes. Collaboration and communication among the digital twins during decision making is conducted in four stages as follows.

At the *first* stage, the cognitive digital twin of an industrial machine (e.g., a grinding machine) equipped with edge analytics is continuously monitoring values against predefined thresholds. An alert is created whenever a threshold is breached (e.g., the temperature of a motor inside the grinding machine goes beyond an acceptable threshold).⁴ At the *second* stage, the cognitive digital twin starts the sensemaking process by collecting contextual information including product characteristics (e.g., to check the rigidity of a metal alloy being used for a product), configurations of the processes being applied by the machine (e.g., pressure and speed of a grinding process), and operational conditions on the factory shop floor such as temperature, humidity, etc. The cognitive digital twins are capable of correlating all acquired information and initiating a sensemaking process to understand whether the current spike in temperature is due to a fault in the machine, characteristics of the

product being manufactured, the manufacturing process being applied, or conditions on the shop floor. A factory level knowledge base is gradually created for all previous anomalies detected and their remedial actions. If a preexisting similar cause is identified, and its remedial action is available in the knowledge base, the cognitive digital twin will adjust its configuration, request a process adjustment, and/or adjust operational conditions accordingly. In the *third* stage, if a cognitive digital twin is unable to make sense of local information, it seeks further assistance from the social network of its peers and requests information from similar machines with similar operational conditions, e.g., a grinding machine of the same make and brand being used in a different plant. If an anomaly in temperature is only being observed locally, the digital twin of the machine adjusts itself to the configuration of machines running optimally without any issues. If the anomaly is observed across the board, a network-wide alert is broadcasted to request remedial actions. In the *fourth* stage, a record of captured events, interactions, the outcome of analytics, and the sensemaking process together with domain expertise is stored in a shared knowledge base in the shape of an enterprise-level knowledge graph. This knowledge graph will act as a central information portal for any future occurrences of similar events. We see that in the future, this knowledge graph will act as a central hub for all operational machines to post questions and get immediate

answers. When necessary, a human expert may also be consulted.

RESEARCH CHALLENGES

To realize the vision of cognitive digital twins, we envision a design and implementation of a distributed cross-domain autonomous system for smart manufacturing. The goal of this system is to enhance autonomous manufacturing by empowering manufacturing resources to think, learn, and understand the dynamics of industrial environments by effectively integrating human cognition through AI and Semantic Web technologies into the design of autonomous manufacturing, respecting the Industry 4.0 system design principles. The approach can be cross-disciplinary, involving AI, semantic-empowered techniques, as well as semantic data integration in autonomous manufacturing scenarios. To achieve the above-mentioned vision, the following intertwined Research Questions (RQs) need to be addressed:

RQ1: HOW TO CREATE AN AUTONOMOUS DISTRIBUTED SYSTEM CONJOINING THE BOTTOM-LEVEL MANUFACTURING RESOURCES TO ENHANCE RESPONSIVENESS AND INTELLIGENCE? THIS RESEARCH QUESTION IS FURTHER DIVIDED INTO THE FOLLOWING RESEARCH AREAS:

- › **A Collaborative Network of Intelligent Agents:**

This research investigates the design of an autonomous system that can discover and detect faults and disturbances autonomously as well as collaboratively. In addition to this, it can attempt to go beyond the existing knowledge of known problems to mitigate new problems and anomalies, thus capable of operating in unknown environments. Furthermore, they can build a collaborative network of intelligent agents locally to improve the responsiveness of the system.

- › **Automated Analytics for Resource-constrained Manufacturing Resources⁸:** This research requires the investigation of the suitability of existing interoperability standards (e.g., Web of Things, RAMI 4.0, Semantic Web) and the suitability of existing architecture patterns

(e.g., fog, Intelligent edge,³ and smart agent) for resource-constrained manufacturing resources as it demands quick response and automatic analytics with enhanced intelligent capabilities.

- › **Autonomous Models on top of Knowledge Graph:**

This research requires investigation of incorporating several autonomous models on top of semantic-empowered technologies as we do not want to limit our vision of cognitive digital twins only for a specific autonomous model. For instance, an integration of self-comparison models, where a single machine can be compared with a fleet of similar machines. This capability can be extended further by leveraging historical information to predict its suitability for autonomous resource allocation.

RQ2: HOW TO ENABLE AN AUTONOMOUS CROSS-DOMAIN REASONING OVER DISTRIBUTED INDUSTRY 4.0 APPLICATIONS?

Industry 4.0 applications are currently designed while keeping a single application domain in view. Most of these applications target a domain-specific problem. Cross-domain collaborations allow to deduce additional events from a silo and can be turned into useful actuation, e.g., before allocating manufacturing resources, a system considers external electricity rates and supply chain data (e.g., weather and traffic conditions) in order to achieve the goal of reducing the factory's energy consumption and carbon footprint.

To address this research question, we need to investigate an autonomous cross-domain system, which can leverage semantic reasoning to derive new knowledge and AI techniques to monitor and process events from totally independent applications. It can integrate the techniques of knowledge discovery and inference that is not possible from data generated by a single application. Moreover, it can use algorithms for autonomous decision-making with uncertain, dynamic, and incomplete information. Having a framework among industrial machines and shared collaborative intelligence identified in RQ1 can prepare the necessary ground to achieve RQ2, synthesizing analytics and

intelligence of factories with other external knowledge and services for decision making. 🌐

ACKNOWLEDGMENTS

This work was supported in part by grants from the European Union's Horizon 2020 research and innovation programme under grant agreement number 847577 (SMART 4.0 Marie Skłodowska-Curie actions COFUND), in part by the Science Foundation Ireland (SFI) under grant number SFI/16/RC/3918 (Confirm) co-funded by the European Regional Development Fund, and in part by SCRA (South Carolina Research Authority).

REFERENCES

1. S. S. Tavallaey and C. Ganz, "Automation to autonomy," in *Proc. 24th IEEE Int. Conf. Emerg. Technol. Factory Autom.*, 2019, pp. 31–34, doi: 10.1109/ETFA.2019.8869329.
2. M. I. Ali, P. Patel, S. K. Datta, and A. Gyrard, "Multi-layer cross domain reasoning over distributed autonomous IoT applications," *Open J. Internet of Things*, vol. 3, no. 1, pp. 75–90, 2017. [Online]. Available: <http://nbn-resolving.de/urn:nbn:de:101:1-2017080613451>
3. P. Patel, M. I. Ali, and A. Sheth, "From raw data to smart manufacturing: AI and semantic web of things for industry 4.0," *IEEE Intell. Syst.*, vol. 33, no. 4, pp. 79–86, Jul./Aug. 2018, doi: 10.1109/MIS.2018.043741325.
4. V. Kamath, J. Morgan, and M. I. Ali, "Industrial IoT and digital twins for a smart factory: An open source toolkit for application design and benchmarking," in *Proc. IEEE Glob. Internet Things Summit*, 2020, pp. 1–6, doi: 10.1109/GIOTS49054.2020.9119497.
5. K. Xia *et al.*, "A digital twin to train deep reinforcement learning agent for smart manufacturing plants: Environment, interfaces and intelligence," *J. Manuf. Syst.*, vol. 58, pp. 210–230, Jan. 2021, doi: 10.1016/j.jmsy.2020.06.012.
6. P. Patel and M. I. Ali, "Developing real-time smart industrial analytics for Industry 4.0 applications," in *Smart Service Management - Design Guidelines and Best Practices*. New York, NY, USA: Springer, 2020, doi: 10.1007/978-3-030-58182-4_17.
7. K. Xia, C. Saidy, M. Kirkpatrick, N. Anumbe, A. Sheth, and R. Harik, "Semantic integration of machine vision systems to aid manufacturing event understanding," submitted for publication.
8. B. Sudarsan, P. Patel, M. I. Ali, J. Breslin, and R. Ranjan, "Towards executing neural networks-based video analytics models on resource-constrained IoT devices," submitted for publication.



IEEE COMPUTER SOCIETY
Call for Papers

Write for the IEEE Computer Society's authoritative computing publications and conferences.

GET PUBLISHED
www.computer.org/cfp

Reversible Execution for Robustness in Embodied AI and Industrial Robots

Ivan Lanese , University of Bologna/INRIA, 40126, Italy

Ulrik P. Schultz , University of Southern Denmark, 5230, Denmark

Irek Ulidowski , University of Leicester, LE1 7RH, U.K.

Reversible computation is a computing paradigm where execution can progress backward as well as in the usual, forward direction. It has found applications in many areas of computer science, such as circuit design, programming languages, simulation, modeling of chemical reactions, debugging, and robotics. In this article, we give an overview of reversible computation focusing on its use in robotics. We present an example of programming industrial robots for assembly operations where we combine classical AI planning with reversibility and embodied AI to increase the robustness and versatility of industrial robots.

Reversibility can be defined as the ability of a program or a system to execute in reverse in order to undo the effects of its (forward) computation. Reversibility has interested scientists for many years. Landauer has discovered over 60 years ago that erasing information in computers requires energy and that loss of information, such as erasing a value stored in a variable, during computation is manifested by the release of heat.¹ The scientists thought at the time that if we could build logic circuits and, ultimately, hardware that reduces or even avoids completely the need to remove information, then computers would be more energy efficient. Subsequently, Fredkin and Toffoli developed reversible universal logic gates as an alternative to the traditional CMOS technology gates.² This meant that, at least in theory, it was possible to design and manufacture reversible computers. There has been a significant amount of research on reversible computers since the discovery of reversible logic gates, culminating in many projects to develop reversible circuits and hardware, but these have not changed the way modern hardware is built yet. Apart from this original

motivation for physical reversibility, there are many other reasons for, and benefits of, logical reversibility.³ The latter form of reversibility concerns enhancing systems and software (that run on a physically irreversible hardware) with the ability to undo (or simulate undoing of) computation. There are reversible programming languages such as Janus⁴ and there are techniques for reversing traditional imperative programming languages such as C.⁵

We have also discovered the basics of how to reverse the computation of concurrent programs and systems.^{6–9}

The purpose of this article is to introduce the topic of reversible computation by presenting a robotics case study where logical reversibility has made a difference. The case study, and more generally, reversible computation research in Europe were partially supported by COST Action IC1405 on Reversible Computation—Extending Horizons of Computing.¹⁰ We shall touch gently on the theories we have developed and explain how they assisted us in solving practical problems of the case study. We will also indicate how we have adjusted our formal techniques to strengthen a traditional AI planning approach to produce a full working solution.

Our case study is about programming industrial robots performing assembly operations (i.e., building a physical product) in a way that, based on a fixed assembly sequence generated by an AI-based planner,

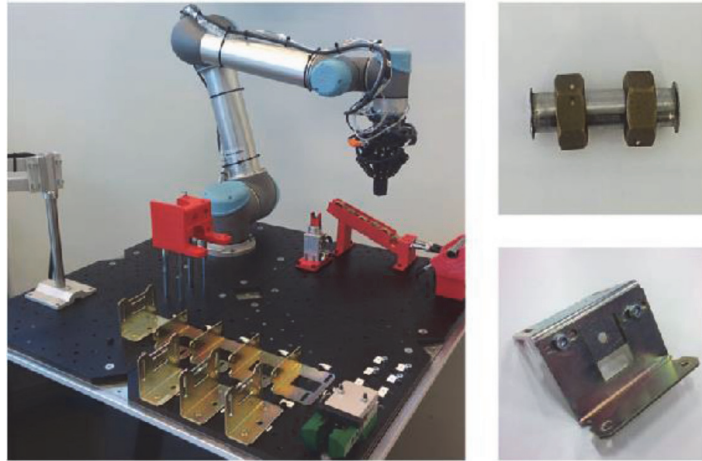


FIGURE 1. Experimental setup: industrial robot (left) and the two product parts being automatically assembled and disassembled (right).

achieves automatic error recovery and even automatic disassembly. Error recovery is achieved by temporarily reversing the direction of execution, effectively undoing recent steps, and then trying again. This approach works well in the physical world of robots because slight imprecisions can cause the robot to get stuck, but partially disassembling the object and trying again can often solve the problem. Taken to an extreme, the entire assembly sequence can be reversed, effectively providing an automatic way to disassemble an object.

We thus demonstrate how a traditional AI-based planning approach is enriched by an underlying reversible execution model that relies on the embodiment of the robot system to provide a robust, probabilistic way of executing the plan. The approach is based on the principles of the Janus reversible programming language,⁴ where every step of the computation must in itself be reversible, thus ensuring that the program as a whole is reversible. In Janus, this means that certain irreversible operations, such as multiplication by zero, are not allowed. Similarly, for the robots, reversible execution can not be applied to intrinsically irreversible steps such as cutting or welding.

REVERSIBILITY IN ROBOTICS

Robots act upon the physical world, and depending on the type of robot, may be capable of performing actions that can be considered reversible.

Consider the specific case of an industrial robot, i.e., a general-purpose robot arm as depicted in Figure 1 (left), normally consisting of six or more joints

connected in series and programmed using a special-purpose robot programming language. Moving an object from one location to another, or screwing two pieces of metal together using a bolt, could be considered reversible actions. Conversely, breaking an object in two or welding two pieces of metal together would not be considered reversible. If a robot is performing a sequence of operations that can be considered reversible, such as the steps required to assemble a kitchen appliance or a photocopier, then could the entire sequence of operations be perhaps considered a reversible program?

This thought experiment motivated the study and development of reversible domain-specific robot programming languages. (Domain-specific languages are special-purpose languages designed to solve specific problems such as robot programming¹¹). The key insight is that if the robot is constrained to only perform operations that are physically reversible, then an entire sequence of operations (i.e., a robot program) can be considered physically reversible. Reversible robot programming languages have been studied for industrial robots and modular self-reconfigurable robots. Industrial robots are the topic of this article.

Self-reconfigurable robots are robots that can physically rebuild themselves to take on different shapes.¹² As a concrete example, the ATRON modular robot¹³ is shown in Figure 2: each module is an individual robot, and a snake robot composed of multiple modules can rebuild itself into a car robot. A modular robot shaped as a car can for example rebuild itself into a snake to traverse an obstacle, and then afterward return to the car shape to continue normal operations.

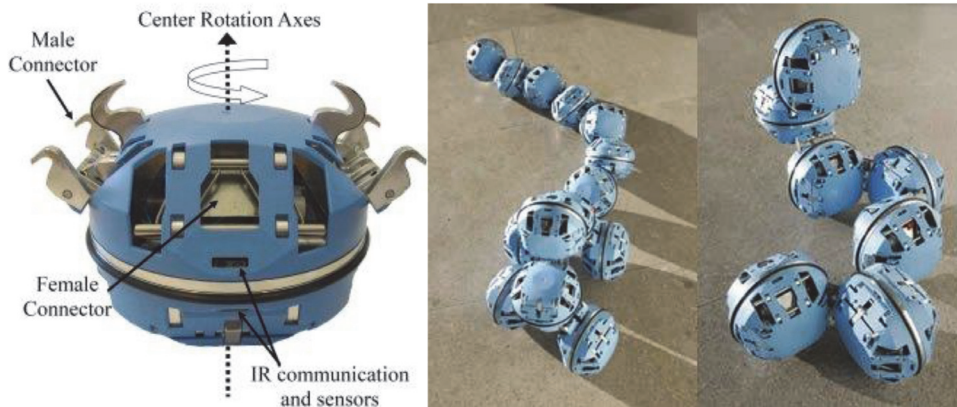


FIGURE 2. ATRON modular robot: a single module (left), car/snake configurations (middle), and the snake changing shape (right).

This process, referred to as self-reconfiguration, can be considered reversible if each of the steps performed by the individual modules is reversible and can be repeated in reverse order.¹⁴

Industrial Robots, AI, and Reversibility

Programming industrial robots is challenging due to the difficulty of precisely specifying general yet robust operations. As the complexity of these operations increases, so does the likelihood of errors. The classical AI-based approach is to derive a plan for the sequence of operations required to complete a given task.^{15, 16}

Such plans however break down in case of errors, resulting in the need to replan the sequence of operations. Replanning can be costly, in particular in complex operations where errors could occur quite often.¹⁷

We propose to generate plans as reversible operation sequences, such that if random failure causes a step to fail, the system can automatically backtrack and retry without replanning. Reverse execution here allows the robot to back out of an erroneous situation, after which the operation can be automatically retried. In a perfectly predictable system retrying would usually result in the same errors, but the embodiment of the robot here works to our advantage: retrying the same physical operation multiple times can produce different results. In effect, the AI-generated plan is made robust toward specific kinds of errors and can be executed robustly without any need for costly dynamic replanning. The combination of automatic retries based on reversibility and probabilistic operations can be considered a form of embodied AI, where reversibly retrying the same operation multiple times results in increased robustness.

As we will see, the combination of AI-based planning and reversibility is amongst others useful for automatic error recovery for small-sized batch production of assembly operations, where precisely specifying error-free operations would be time-consuming and expensive, and dynamic replanning would add a significant overhead to the operation. Moreover, reversibility can in this case be used to automatically derive a disassembly sequence from a given assembly sequence or vice versa. These capabilities can be achieved using a reversible domain-specific language for specifying assembly sequences.

Robotic assembly and disassembly are done in terms of sequences of operations, such as placement of objects, insertions, screwing operations, and so forth. All are challenged by uncertainties from sensors, robot kinematics, and part tolerances. Not all operations are reversible, some are not even repeatable! Many physical phenomena and actions can nevertheless be considered reversible, depending on the abstraction level at which they are observed. For example, an industrial robot that pushes an object to a new position could easily move this object back to its original position, but cannot simply do this by reversing its pushing movements, as pulling requires gripping the object first. Moreover, some operations, such as cutting and welding, are in practice nonreversible. A study of 13 real-world industrial cases showed roughly 76% of the operations to be reversible,¹⁸ but many of the operations require the robot to perform a different physical action to reverse a given action. Taking inspiration from this study, we divide reversible operations into two categories: directly reversible operations that simply can be reversed by performing the forward action in reverse; and indirectly reversible operations, which can be reversed, but require a


```
// SCREWING OPERATION
sequence("insert_screw_operation").
  action(insert_screw).
    reverseWith("remove_screw").nonreversible().
  call("insert_screw_suboperation").
  move(qScrewInside).
  move(qScrewOutside);
```

FIGURE 3. Sample reversible assembly program: a sequence is defined to consist of an “insert screw” action, a call to another sequence, and two move actions. The “insert screw” action is not itself reversible and is marked as indirectly reversible using the “remove screw” action.

different sequence of instructions, which must be manually specified by the programmer.

In the design of our robot programming language, we take inspiration from the Janus reversible programming language, where programs are said to be time-invertible.⁴ Each computational step in Janus has a specific inverse, and a given program that when executed forwards computes a function will compute the inverse of this function when executed backward. Subtracting a constant is for example the inverse of adding a constant. In our robot programming language, each physically reversible operation similarly has an inverse. In the case of directly reversible operations, the inverse is automatically derived by the system. In the case of indirectly reversible operations, the programmer must manually specify the sequence of operations that constitutes the reverse of a given operation. Executing such a manually specified inverse can temporarily bring the system into a state not normally encountered during forward execution. Moreover, switching execution direction in the middle of such a manually specified inverse might again take the system into a new state. This contrasts a main property of reversibility in programming languages like Janus and of causal-consistent reversibility,^{6, 7} a notion used in concurrent systems, which says that any reachable state is forwards reachable. In causal-consistent reversibility, any step of computation of a concurrent system can be undone provided that all its effects, if any, are undone first.⁹ Such a property also fails in other contexts, e.g., in some biological systems.¹⁹ Since an operation and its (indirect) reverse are paired, the program has unique starting and ending states, and execution will only terminate in one of these two states. Infinite loops of error correction can manifest and are handled using a monitoring heuristic that detects if the assembly operation might be stuck.

The programming model we have developed is based on this abstract semantics-based model extended with various features required for reversible control of industrial robots in real-world scenarios.¹⁸

The actual implementation is in the form of an internal DSL in C++, meaning that a sequence of C++ method calls is used to build a model of the reversible assembly sequence, as shown in Figure 3. A robot assembly task is programmed as a sequential flow of operations. It is sequential since in practice assembly tasks tend to be a simple sequence of operations (except for error handling, but we aim to automatically handle errors using reverse execution). Reversibility is nevertheless still relevant due to the presence of random behavior of the physical operations: reversing and re-executing an operation may produce a different result. Each operation (denoted by the keyword “sequence”) represents a high-level assembly case logic and is a sequence of primitive instructions. Each instruction is either directly reversible (default), indirectly reversible (indicated by the keyword “reverseWith”), or nonreversible (indicated by “nonreversible”).

Our approach was evaluated experimentally using two industrial assembly use-cases,¹⁸ Figure 1 shows the physical robot platform (left) and the two assembled use-cases (right). Both use-cases were used to test the principle of reversible assembly and the use of reverse execution for error correction. For reversible assembly, the program was executed forward to assemble each use-case. Afterward the finished object was manually placed back into the system, and the program was executed backward to disassemble the object. This was done multiple times for each use-case with no errors.

The use of reverse execution as an effective error correction tool was experimentally demonstrated by assembling a large number of objects, as follows. The workcell was set to assemble 100 objects of each type consecutively and without pause. During these 200 assemblies a total of 22 errors occurred, of which 18, corresponding to 82%, were automatically resolved and corrected using reverse execution. Errors that were automatically corrected include failed peg-in-hole operations (fixed by backtracking and trying

again), dropping a tube (fixed by reversing until a new tube was picked from the feeder), failed to grasp a screw, and screwing failing due to misalignment.

Errors that could not be automatically corrected include air-tubing from the gripper getting stuck on the platform, causing the gripper to misalign, and a screw being inserted at a skewed angle causing a bracket to misalign, which could not be corrected as the system had no means of detecting the bracket misalignment.

CONCLUSION

Reversing of computation is conceptually and technically a challenging task even if we only consider logical reversibility. We have illustrated significant potential benefits of reversibility to improve AI-planning in the robotics case study.

We have presented briefly some of the recently developed theoretical underpinnings for the case study, concentrating mainly on explaining how reversibility helps. Exploring this application area helped us to exemplify the richness of different forms of reversibility.

While the case study we have discussed is based on sequential reversibility, the notion of causal-consistent reversibility,^{6, 7} is key to scalable reversible programming of modular robotic systems such as the ATRON robot shown in Figure 2. This is because the modules perform operations in parallel and hence reversing the system must respect dependencies between the actions of individual modules. Provided the ATRON robot does not perform any irreversible steps, we have this strong property: any reachable (by an arbitrary combination of reverse and forward steps) state is forwards reachable. Applying causal consistency to achieve scalable and robust reversible programming for swarming robot systems like the ATRON and unmanned aerial vehicles (UAVs, drones) is considered future work. In the specific case of UAVs, a programming model based on reversibility would hypothetically allow a drone swarm as a whole to “reverse” distributed control decisions, thus easing requirements on reaching consensus (before beginning new operations) and increasing the robustness of the system.

Robots are controlled through programming, but we cannot be certain of their actions since they interact with an unpredictable physical world. Contrary to causal-consistent reversibility, we have seen that some inverses of indirectly reversible operations may lead to new “get-out-of-trouble” states, albeit temporarily, which are not forwards reachable. Such states are needed due to the irreversibility of the physical world with which the robot interacts.

There are also other forms of reversibility suitable for different applications. Probably the best known is backtracking, where steps of computation are undone in the inverse order of execution.

Apart from many traditional applications of backtracking such as, for example, in search algorithms or logic programming, it has been used to undo concurrent C-like programs for debugging.^{10, 20} 🌐

ACKNOWLEDGMENTS

The authors acknowledge partial support from COST Action IC1405 on Reversible Computation—Extending Horizons of Computing. The work of I. Lanese was supported in part by French ANR project DCore ANR-18-CE25-0007.

REFERENCES

1. R. Landauer, “Irreversibility and heat generated in the computing process,” *IBM J. Res. Develop.*, vol. 5, pp. 183–191, 1961, doi: 10.1147/rd.53.0183.
2. E. Fredkin and T. Toffoli, “Conservative logic,” *Int. J. Theor. Phys.*, vol. 21, pp. 219–253, 1982, doi: 10.1007/BF01857727.
3. C. Bennett, “Logical reversibility of computation,” *IBM J. Res. Develop.*, vol. 17, pp. 525–532, 1973, doi: 10.1147/rd.176.0525.
4. T. Yokoyama, H. B. Axelsen, and R. Glück, “Principles of a reversible programming language,” in *Proc. Comput. Frontiers*, 2008, pp. 43–54, doi: 10.1145/1366230.1366239.
5. K. Perumalla, *Introduction to Reversible Computing*. Boca Raton, FL, USA: CRC Press, 2014, doi: ISBN 9781439873403.
6. V. Danos and J. Krivine, “Reversible communicating systems,” in *Proc. CONCUR*, 2004, vol. 3170, pp. 292–307, doi: 10.1007/978-3-540-28644-8_19.
7. I. Phillips and I. Ulidowski, “Reversing algebraic process calculi,” *J. Logic Algebr. Program.*, vol. 73, pp. 70–96, 2007, doi: 10.1016/j.jlap.2006.11.002.
8. I. Lanese, C. Mezzina, and J.-B. Stefani, “Reversing higher-order pi,” in *Proc. CONCUR*, 2010, vol. 6269, pp. 478–493, doi: 10.1007/978-3-642-15375-4_33.
9. I. Lanese, I. Phillips, and I. Ulidowski, “An axiomatics approach to reversible computation,” in *Proc. FOSSACS*, 2020, vol. 12077, pp. 442–461, doi: 10.1007/978-3-030-45231-5_23.
10. I. Ulidowski, I. Lanese, U. P. Schultz, and C. Ferreira, eds., *Reversible Computation: Extending Horizons of Computing - Selected Results of the COST Action IC1405, ser. LNCS*. Berlin, Germany: Springer, 2020, vol. 12070, doi: 10.1007/978-3-030-47361-7.

11. M. Fowler, *Domain-Specific Languages*. Boston, MA, USA: Addison-Wesley, 2010, doi: 10.5555/1809745.
12. M. Yim et al., "Modular self-reconfigurable robot systems [Grand challenges of robotics]," *IEEE Robot. Automat. Mag.*, vol. 14, no. 1, pp. 43–52, Mar. 2007, doi: 10.1109/MRA.2007.339623.
13. M. W. Jorgensen, E. H. Ostergaard, and H. H. Lund, "Modular ATRON: Modules for a self-reconfigurable robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2004, pp. 2068–2073, doi: 10.1109/IROS.2004.1389702.
14. U. Schultz, M. Bordignon, and K. Stoy, "Robust and reversible execution of self-reconfiguration sequences," *Robotica*, vol. 29, pp. 35–57, 2011, doi: 10.1017/S0263574710000664.
15. R. E. Fikes and N. J. Nilsson, "Strips: A new approach to the application of theorem proving to problem solving," in *Artif. Intell.*, vol. 2, no. 3/4, 1971, pp. 189–208, doi: 10.1016/0004-3702(71)90010-5.
16. C. R. Garrett, T. Lozano-Pérez, and L. P. Kaelbling, "Backward-forward search for manipulation planning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 6366–6373, doi: 10.1109/IROS.2015.7354287.
17. P. Loborg, "Error recovery supporting manufacturing control systems," Ph.D. dissertation, School Eng., Linköping Univ., Linköping, Sweden, <http://urn.kb.se/resolve?urn=urn%3Anbn%3Ase%3Aliu%3Adiva-163284>, 1994, doi: ISBN 9178713765.
18. J. Laursen, L. Ellekilde, and U. Schultz, "Modelling reversible execution of robotic assembly," *Robotica*, vol. 36, no. 5, pp. 625–654, 2018, doi: 10.1017/S0263574717000613.
19. I. Phillips, I. Ulidowski, and S. Yuen, "A reversible process calculus and the modelling of the ERK signalling pathway," in *Proc. Reversible Comput.*, 2013, vol. 7581, pp. 218–232, doi: 10.1007/978-3-642-36315-3_18.
20. J. Hoey and I. Ulidowski, "Reversing imperative parallel programs and debugging," in *Proc. Reversible Comput.*, 2019, vol. 11497, pp. 108–127, doi: 10.1007/978-3-030-21500-2_7.

IVAN LANESE is an Associate Professor with the University of Bologna, Bologna, Italy. He acted as the Vice Chair of the COST Action IC1405 on Reversible Computation. He is interested in formal methods and concurrency theory, with special focus on reversibility and reversible debugging. Further information is available at <https://www.unibo.it/sitoweb/ivan.lanese/en>. He can be contacted at ivan.lanese@gmail.com.

ULRIK P. SCHULTZ is the Professor in aerial robotics with the University of Southern Denmark, Odense, Denmark. He recently participated in the COST Action IC1405 on reversible computing where he chaired the Working Group on Applications. He is interested in programming languages for robotics, his full biography is available at <http://www.sdu.dk/staff/ups> and he can be contacted at ups@sdu.dk.

IREK ULIDOWSKI is an Associate Professor with the University of Leicester, Leicester, U.K. He chaired the COST Action IC1405 on reversible computation. His interests include reversible computation and its application, and concurrent systems. Further information is available at <https://www.cs.le.ac.uk/people/iu3>. He can be contacted at irekulidowski@gmail.com.

A Brief History of Warehouse-Scale Computing

Reflections Upon Receiving the 2020 Eckert-Mauchly Award

Luiz André Barroso , Google, Mountain View, CA, 94043, USA

Receiving the 2020 ACM-IEEE Eckert-Mauchly Award this past June was among the most rewarding experiences of my career. I am grateful to *IEEE Micro* for giving me the opportunity to share here the story behind the work that led to this award, a short version of my professional journey so far, as well as a few things I learned along the way.*

THE PRACTICE OF COMPUTER SCIENCE

For many of us our earliest models of professionalism come from observing our parents' approach to their work. That was the case for me observing my father, a surgeon working in public hospitals in Rio de Janeiro. Throughout his career, he was continually investigating new treatments, collecting case studies, participating and publishing in medical conferences, despite never having held an academic or research position. He was dedicated to the practice of medicine but always made time to help advance knowledge in his area of expertise.

Without really being aware of it, I ended up following my father's path and became a practitioner myself. As a practitioner, my list of peer-reviewed publications is notably shorter than most of the previous winners of this award, but every time I had something valuable to share with the academic community, I felt welcomed by our top research conferences, and those articles tended to be well received. Practitioners like myself tend to publish papers in the past tense, reporting on ideas that

have been implemented and launched as products. Practitioners can contribute to our community by looking back and showing us how those ideas played out (or not) in practical applications. Commercial success or the lack thereof can be an objective judge of the merits of research ideas; even if cruelly so at times. In giving me this award, the IEEE Computer Society and ACM are highlighting the role of practitioners in our field.

Now, as this award is about the practice of warehouse-scale computing, I should get to that with no further delay.

A BRIEF HISTORY OF WAREHOUSE-SCALE COMPUTING

If it is indeed true that "great poets imitate and improve,"¹ poetry and computing may have something in common after all. Warehouse-scale computers (the name we eventually gave to the computers we began to design at Google in the early 2000s) are the technical descendents of numerous distributed computing systems that aimed to make multiple independent computers behave as a single unit. That family begins with VAXclusters² in the 1980s, a networked collection of VAX computers with a distributed lock manager that attempted to present itself as a single system to the user. In the 1990s, the concept of computing clusters began to be explored using lower end or desktop computers and local area networks with systems such as NASA's Beowulf clusters³ and UC Berkeley's NOW project.⁴

*Administered jointly by ACM and the IEEE Computer Society, the award is given for contributions to computer and digital systems. In 2020, my award was given for pioneering the design of warehouse-scale computing and driving it from concept to industry.

*FOR MANY OF US OUR EARLIEST
MODELS OF PROFESSIONALISM
COME FROM OBSERVING OUR
PARENTS' APPROACH TO THEIR
WORK. THAT WAS THE CASE FOR ME
OBSERVING MY FATHER, A SURGEON
WORKING IN PUBLIC HOSPITALS IN
RIO DE JANEIRO.*

When I arrived at Google, in 2001, I found a company of brilliant programmers that was short on cash but not on confidence as they had already committed to a strategy of systems built from inexpensive desktop-class components. Cheap might be a fairer characterization of those early systems than inexpensive. The first generation of those computer racks, tenderly nicknamed “corkboards” consisted of desktop motherboards loosely resting on sheets of cork that isolated the printed circuit boards from the metal tray, with disk drives themselves loosely resting on top of DIMMs.

Despite my hardware background,[†] I had joined Google to try to become a software engineer. In my early years, I was not involved in building computers but instead I worked developing our index searching software and related software infrastructure components such as load balancers and remote procedure call libraries. Three years later, Urs Hölzle asked me to build a hardware team capable not only of building sound server-class systems but to invent new technologies in the datacenter space. The years I had spent in software development turned out to be extremely useful in this new role since my first-hand understanding of Google’s software stack was essential to architecting the machinery needed to run it. We published some of those early insights into the architectural requirements for Google-class workloads in an *IEEE Micro* paper in 2003.⁶

OUR TEAM’S LACK OF EXPERIENCE IN DATACENTER DESIGN MAY HAVE BEEN AN ASSET AS WE SET OUT TO QUESTION NEARLY EVERY ASPECT OF HOW THESE FACILITIES WERE DESIGNED

In our earliest days as a hardware team we focused primarily on designing servers and datacenter networking, but quickly realized that we would need to design the datacenters themselves. Up until that point internet companies deployed computing machinery in third-party colocation facilities (businesses that provisioned space, power, cooling, and internet connectivity for large scale computing gear), and Google was no exception. As the scale of our deployments grew, the minimum footprint required for a Google cluster was beginning to be larger than the total size of existing

co-location facilities, so we had to build our own facilities in order to continue to grow our services.

At that point, it became evident to us how much room for improvement there was in the design of datacenters. As a third-party hosting business, datacenters were put together by groups of disjoint engineering crafts that knew little of each other’s disciplines; civil engineers built the building, mechanical engineers provisioned cooling, electrical engineers distributed power, hardware designers built servers, software engineers wrote internet services. The lack of cross-disciplinary coordination resulted in facilities that were both expensive and incredibly energy inefficient. Our team’s lack of experience in datacenter design may have been an asset as we set out to question nearly every aspect of how these facilities were designed. Perhaps most importantly we had the chance to look at the entire system design, from cooling towers to compilers, and that perspective quickly revealed significant opportunities for improvement.

Speed of deployment was also a critical factor in those days as we were often running dangerously close to exhausting our computing capacity as our traffic grew, so our initial approach was to prefabricate ready-to-deploy computer rooms inside forty foot shipping containers. Containers gave us a datacenter floor where we could isolate the hot (exhaust) plenum from the cold aisle and shortened the total distance the air needed to be moved; both were factors that improved cooling efficiency. All that the container needed to function was power, cold water and networking, and we had a 1200-server machine room ready to deploy.

That original container-based deployment also introduced other innovations that led to cost, performance, and energy efficiency improvements. Here are some of the most notable ones:

- *Higher temperature air cooling:* We determined through field experiments that contrary to common wisdom the electronic components believed to be most affected by air temperature were still quite reliable at reasonably high temperatures (think 70F instead of 60F).⁸ This made it possible to run many facilities using evaporative cooling and improved cooling efficiency.
- *Distributed uninterruptible power supplies (UPS):* Typical datacenters were built with a UPS room (a room full of batteries) in order to store enough energy to ride electrical grid glitches. As such ac voltage was rectified to power the UPS and then inverted to distribute to the machine room only then to be rectified again by per-server power supplies, incurring losses at each transformation

[†]My Ph.D. and the earlier phase of my career had been in computer architecture, particularly in microprocessor and memory system design.



FIGURE 1. A Google warehouse-scale computer in Belgium.

step. We instead eliminated the UPS room and introduced per tray (and later per rack) batteries. That way power entering the building only needed to be rectified once per machine.

- › *Single-voltage rail power supplies:* Every server used to be outfitted with a power supply that converted ac voltage into a number of dc voltage rails ($\pm 12\text{ V}$, $\pm 5\text{ V} \pm 3.3\text{ V}$, etc.) based on old standards for electronic components. By 2005, most electronic components did not use any of the standard dc rails so yet another set of dc/dc conversions needed to happen onboard. The allocation of power among multiple rails also lowered power supply efficiency sometimes below 70%. We introduced a single-rail power supply that reached 90% efficiency and created on-board only the voltages actually used by components.
- › *1000-port GigE Ethernet switch:* Datacenter networking bandwidth was beginning to become a bottleneck for many warehouse-scale applications, but enterprise-grade switches were not only very expensive but also lacked offerings for large numbers of high bandwidth endpoints. Using a collection of inexpensive edge switches configured as a multistage network, our team created the first of a family of distributed datacenter networking products (codenamed Firehose) that could deliver

a gigabit of nonoversubscribed bandwidth to up to a thousand servers.

Although our adventure with shipping containers lasted only that one generation and soon after we found ways to obtain the same efficiencies with more traditional building shells, the innovations from that first program have continued to evolve into industry-leading solutions over generations of warehouse-scale machines. Figure 1 shows a birds-eye view of a modern Warehouse-scale computer.

MY JOURNEY

I knew I wanted to be an electrical engineer when I was 8 years old and got to help my grandfather work on his HAM radio equipment. Putting aside the fact that eight-year-olds should not be making career choices, I find it difficult to question that decision to this date. Although I had always been a good student, I struggled a bit during my Ph.D. and graduated late. I did have a few things going for me: an ability to focus, stamina for hard work, and a lot of luck. As an example, after a 24-year drought the Brazilian men's national soccer team chose to win a World Cup, during my hardest year in graduate school, delivering a degree of joy that was badly needed to get me to the finish line. Less than a year after that World Cup I was working in my grad student office on a

Saturday afternoon when I got a call from Norm Jouppi inviting me to interview for a research job at Digital Equipment's Western Research Lab (WRL). At the time Norm was already one of the most highly respected computer architects in the world and perhaps nothing in my career since has compared to the feeling I had that day—Norm Jouppi knew who I was!

I KNEW I WANTED TO BE AN ELECTRICAL ENGINEER WHEN I WAS 8 YEARS OLD AND GOT TO HELP MY GRANDFATHER WORK ON HIS HAM RADIO EQUIPMENT. PUTTING ASIDE THE FACT THAT EIGHT-YEAR- OLDS SHOULD NOT BE MAKING CAREER CHOICES, I FIND IT DIFFICULT TO QUESTION THAT DECISION TO THIS DATE.

I joined DEC WRL and had the chance to learn from top researchers like Kourosh Gharachorloo and collaborate with leading computer architects such as Sarita Adve, Susan Eggers, Mateo Valero, and Josep Lariba-Pey. During that time, I also met Mark Hill who would become a friend and a mentor. Later, at Google I would also have the chance to coauthor papers with other leading figures in our field such as Tom Wenisch, Wolf Weber, David Patterson, and Christos Kozyrakis.

Perhaps nothing summarizes the impact that friends and luck can have in your life more than the story of how I came to join Google. As I was trying to make a decision between two options, Jeff Dean asked me whether the other company I was considering had also served me crème brûlée during my interviews. I thanked Jeff and accepted the Google offer the very next morning.

The brilliance and generosity of countless people at Google have been essential to the work that led to this award, but I must highlight three here: Urs Hölzle who has been a close collaborator and possibly the single person most to blame for Google's overall systems infrastructure successes; Bart Sano who managed the Platforms team that built out the infrastructure we have today (I was the technical lead for Bart's team for many years); and Partha Ranganathan who is our computing technical lead today and is taking Google's architectural innovation into the future.

One part of my career I have no hesitation to brag about is the quality of the students I have had a chance to host as interns at DEC and Google. They were (to date) Partha Ranganathan, Rob Stets, Jack

Lo, Sujay Parekh, Ed Bugnion, Alex Ramirez, Gautham Thambidorai, Karthik Sankaranarayanan, David Meisner, and David Lo. We worked together on many fun projects and I hope for more in the future. Although my dad is no longer with us, I am also fortunate to count on the love and support of my family. My mom Cecilia, my godmother Margarida, my siblings Paula, Tina, and Carlos and their families, and my wife Catherine Warner who is the award life gives me every single day.

PERHAPS NOTHING SUMMARIZES THE IMPACT THAT FRIENDS AND LUCK CAN HAVE IN YOUR LIFE MORE THAN THE STORY OF HOW I CAME TO JOIN GOOGLE. AS I WAS TRYING TO MAKE A DECISION BETWEEN TWO OPTIONS, JEFF DEAN ASKED ME WHETHER THE OTHER COMPANY I WAS CONSIDERING HAD ALSO SERVED ME CRÈME BRÛLÉE DURING MY INTERVIEWS. I THANKED JEFF AND ACCEPTED THE GOOGLE OFFER THE VERY NEXT MORNING.

THREE LESSONS

I will finish this essay by sharing with you three lessons I have learned in this first half of my career, in the hope that they may be useful to engineers who are at an earlier stage in their journey.

Consider the Winding Road

As an engineer you stand on a foundation of knowledge that enables you to branch into many different kinds of work. Although there is always risk when you take on something new, the upside of being adventurous with your career can be amazingly rewarding. I for one never let my complete ignorance about a new field stop me from giving it a go.

As a result, I have worked in areas ranging from chip design to datacenter design; from writing software for web search to witnessing my team launch satellites into space; from writing software for Google Scholar to using ML to automatically update Google Maps; from research in compiler optimizations to deploying exposure notification technology to curb the spread of Covid-19.⁸

It seems a bit crazy, but not going in a straight line has worked out really well for me and resulted in a rich

set of professional experiences. Whatever the outcome, you will be inoculated from boredom.

Develop Respect for the Obvious

The surest way to waste a career is to work on unimportant things. I have found that big, important problems have one feature in common: they tend to be straightforward to grasp even if they are hard to solve. Those problems stare you right in the face. They are obvious and they deserve your attention.

Let me give you some examples by listing some of my more well-cited papers next to the formulation of the problems address:

Publication	Problem addressed
ISCA'98: "Memory System Characterization of Commercial Workloads" ¹⁰ with Kourosh Gharachorloo and Edouard Bugnion	"High-end microprocessors are being sold to run commercial workloads, so why are we designing them for number crunching?"
ISCA'00: "Piranha: A Scalable Architecture Based on Single-Chip Multiprocessing" ⁵ with Kourosh Gharachorloo, Robert McNamara, Andreas Nowatzky, Shaz Qadeer, Barton Sano, Scott Smith, Robert Stets, and Ben Verghese	"Thread-level parallelism is easy. Instruction level parallelism is hard. Should we bet on thread-level parallelism then?"
CACM '17: "The Attack of the Killer Microsecond" ¹¹ with Mike Marty, Dave Patterson, and Partha Ranganathan	"If datacenter-wide events run at microsecond speeds, why do we only optimize for millisecond and nanosecond latencies?"
CACM '13: "The Tail at Scale" ¹² with Jeff Dean	"Large scale services should be resilient to performance hiccups in any of their subcomponents"
IEEE Computer '07: "A Case for Energy-proportional Computing" ¹³ with Urs Hölzle	"Shouldn't servers use little energy when they are doing little work?"

If it takes you much more than a couple of sentences to explain the problem you are trying to solve, you should seriously consider the possibility of it not being that important to be solved.

Even Successes Have a "Sell-By" Date

Some of the most intellectually stimulating moments in my career have come about when I was forced to

revisit my position on technical matters that I had invested significant time and effort on, especially when the original position had a track record of success. I will present just one illustrative example.

I JOINED GOOGLE AFTER A FAILED MULTIYEAR CHIP DESIGN PROJECT AND AS SUCH I IMMEDIATELY EMBRACED GOOGLE'S DESIGN PHILOSOPHY OF STAYING AWAY FROM SILICON DESIGN OURSELVES.

I joined Google after a failed multiyear chip design project and as such I immediately embraced Google's design philosophy of staying away from silicon design ourselves. Later as the technical lead of Google's data-center infrastructure, I consistently avoided using exotic or specialized silicon even when they could demonstrate performance of efficiency improvements for some workloads, since betting on the low cost base of general purpose components consistently proved to be the winning choice. Year after year, betting on general purpose solutions proved successful.

Then, deep learning acceleration for large ML models arose as the first opportunity in my career to build specialized components that would have both broad applicability and dramatic efficiency advantages when compared to general purpose designs. Our estimates indicated that large fractions of Google's emerging AI workloads could be executed in these specialized accelerators with as much as a 40× cost/efficiency advantage over general purpose computing.

That was a time to ignore the past successes of betting on general purpose off-the-shelf components and invest heavily on the design and deployment of our own silicon to accelerate ML workloads. Coming full circle, this meant that it was now my time to call Norm Jouppi and ask him to join us to become the lead architect for what was to become our TPU accelerators program.

CONCLUDING

Before the onset of the current pandemic, some of us may have underappreciated how important computing technology and cloud-based services have become to our society. In this last year, these technologies have allowed many of us to continue to work, to connect with loved ones, and to support each other. I am grateful to all of those at Google and everywhere in our industry who have built such essential technologies, and I am inspired to be working in a field with still so much potential to improve people's lives. 🌍

REFERENCES

1. W. H. Davenport Adams, "Imitators and plagiarists," *The Gentleman's Magazine*, Jan. 1892
2. N. P. Kronenberg, H. M. Levy, and W. D. Strecker, "VAXcluster: A closely-coupled distributed system," *ACM Trans. Comput. Syst.*, vol. 4, May 1986, Art. no. 130. [Online]. Available: <https://doi.org/10.1145/214419.214421>
3. T. Sterling, D. Becker, M. Warren, T. Cwik, J. Salmon, and B. Nitzberg, "An assessment of Beowulf class computing for NASA requirements: Initial findings from the first NASA workshop on Beowulf-class clustered computing," in *Proc. IEEE Aerosp. Conf.*, 1998, pp. 367–381.
4. T. E. Anderson, D. E. Culler, and D. Patterson, "A case for NOW (Networks of Workstations)," *IEEE Micro*, vol. 15, no. 1, pp. 54–64, Feb. 1995.
5. L. A. Barroso *et al.*, "Piranha: A scalable architecture based on single-chip multiprocessing," in *Proc. 27th Annu. Int. Symp. Comput. Archit.*, 2000, pp. 282–293.
6. L. A. Barroso, J. Dean, and U. Holzle, "Web search for a planet: The Google cluster architecture," *IEEE Micro*, vol. 23, no. 2, pp. 22–28, Mar./Apr. 2003.
7. A. Singh *et al.*, "Jupiter rising: A decade of Clos topologies and centralized control in Google's datacenter network," *SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, pp. 183–197, Oct. 2015.
8. E. Pinheiro, W. Weber, and L. Barroso, "Failure trends in a large disk drive population," in *Proc. 5th USENIX Conf. File Storage Technol.*, Feb. 2007, pp. 17–29.
9. Google & Apple Exposure Notification technology. 2020. [Online]. Available: g.co/ENS
10. L. A. Barroso, K. Gharachorloo, and E. Bugnion, "Memory system characterization of commercial workloads," *SIGARCH Comput. Archit. News*, vol. 26, no. 3, pp. 3–14, Jun. 1998.
11. L. Barroso, M. Marty, D. Patterson, and P. Ranganathan, "Attack of the killer microseconds," *Commun. ACM*, vol. 60, no. 4, pp. 48–54, Apr. 2017.
12. J. Dean and L. A. Barroso, "The tail at scale," *Commun. ACM*, vol. 56, no. 2, pp. 74–80, Feb. 2013.
13. L. A. Barroso and U. Holzle, "The case for energy-proportional computing," *Computer*, vol. 40, no. 12, pp. 33–37, Dec. 2007.

LUIZ ANDRÉ BARROSO is a Google Fellow and a former VP of Engineering at Google. His technical interests include machine learning infrastructure, privacy, and the design and programming of warehouse-scale computers. He has published several technical papers and has co-authored the book *The Datacenter as a Computer*, now in its 3rd edition. He is a Fellow of the ACM and the AAAS and he is a member of the National Academy of Engineering. Barroso received a B.S. and an M.S. in electrical engineering from the Pontificia Universidade Católica of Rio de Janeiro, Rio de Janeiro, Brazil, and a Ph.D. in computer engineering from the University of Southern California, Los Angeles, CA, USA. He is the recipient of the 2020 Eckert-Mauchly award. Contact him at luiz@barroso.org.



CALL FOR ARTICLES

IT Professional seeks original submissions on technology solutions for the enterprise. Topics include

- emerging technologies,
- cloud computing,
- Web 2.0 and services,
- cybersecurity,
- mobile computing,
- green IT,
- RFID,
- social software,
- data management and mining,
- systems integration,
- communication networks,
- datacenter operations,
- IT asset management, and
- health information technology.

We welcome articles accompanied by web-based demos.

For more information, see our author guidelines at www.computer.org/itpro/author.htm.

WWW.COMPUTER.ORG/ITPRO



Software Engineering: A Profession in Waiting

David Lorge Parnas, *Middle Road Software, Inc.*

Ongoing efforts to make software development an engineering discipline will fail until we have legislation requiring that creators of certain types of software be licensed, establishing a licensing authority, and detailing the capabilities that a licensed developer must possess.

For more than 50 years, people concerned about software development have tried to make it a profession like medicine or civil engineering.¹ Those who started discussing “software engineering” in the 1960s had been trained in other fields; their work required computation and that led them to writing programs—often for their own use. A few were writing programs that would be used by others; such programs became known as *software*. Some of them observed that they were doing something very different from what they had been taught to do. They had been educated as mathematicians or scientists. Software development was more like engineering. They had been trained to extend knowledge but were now applying that knowledge to build products. That thought suggested the term *software engineering*. Those who introduced that term hoped that properly educated software developers would produce trustworthy products and thereby earn the good reputation enjoyed by professional engineers. Most of them would agree that we have not succeeded.

Traditional professions, such as engineering, law, and medicine, have

- ▶ a licensing authority, which identifies and certifies individuals who are competent to practice the profession and takes action against practitioners who either lack the required capabilities or do not practice properly
- ▶ demand-side legislation, which states that certain services and products can only be provided by licensed professionals
- ▶ an accreditation authority, which reviews educational programs and approves those that teach the capabilities required for licensing.

Licensed professionals usually display a certificate showing that they are licensed; they may also display their diploma(s) from accredited higher education institutions.

PROFESSIONAL ENGINEERING: AN ABBREVIATED HISTORY

Licensing authorities were introduced when it became clear that people or organizations that need engineering services might not be able to distinguish individuals who were qualified to do the job from others offering to perform that service.

Demand-side legislation was introduced because legislators became aware that the public could be endangered when a service provider or product designer was not qualified to do the job.

Accreditation authorities were created because



the licensing authorities, realizing that the education of an applicant for a license is a key determinant of the applicant's capabilities, found it more efficient to evaluate programs than to evaluate the education of each applicant. Accreditation also helps prospective students to pick a program that is likely to bring them closer to their career goals.

BODY OF KNOWLEDGE OR BODY OF CAPABILITIES?

Many of the efforts to establish a software engineering profession have proposed a "body of knowledge" (BoK) for that field. A BoK is useful for characterizing a science because a science is an organized body of knowledge. Engineering is different. As Theodore von Kármán, a famous Hungarian-American mathematician and aerospace engineer, said, "Scientists

FOR MORE THAN 50 YEARS, PEOPLE CONCERNED ABOUT SOFTWARE DEVELOPMENT HAVE TRIED TO MAKE IT A PROFESSION LIKE MEDICINE OR CIVIL ENGINEERING.

discover the world that exists; engineers create the world that never was." Engineering requires knowledge, but it also requires the ability to apply that knowledge when designing. Engineering and other professions are better characterized by the capabilities required of their practitioners than by the required knowledge.

Efforts to identify a BoK for software engineering seem doomed to fail for several reasons.

- › The set of concepts and facts known to software developers is huge; there is little agreement on the importance and usefulness of even

the most popular concepts. The size of the BoK has caused some developers to try to prioritize the knowledge and identify a required subset, but that is very difficult. What seems important to some seems useless to others.

- › The software BoK is always rapidly growing; further, much of it becomes irrelevant just a few years after it has been added.
- › Much of that knowledge is tied to specific tools. The characteristics of those tools are the result of many arbitrary design decisions, and the tools may evolve or become out of date; consequently, some of the knowledge about those tools will not be of lasting value.

To make sure that the graduates can have lengthy productive careers, the capabilities taught must be fundamental and of lasting value. There is a small set of basic principles and concepts that can be applied by software developers, but they are abstract and often hard to apply. The ability to use those principles must be taught; that is why the field is better defined by a body of capabilities rather than by a BoK. An attempt to identify the capabilities required of software engineers can be found in the article by Landwehr et al.³ These capabilities include much more than the ability to write programs well.²

SOFTWARE DEVELOPMENT IS NOT NOW AN ENGINEERING PROFESSION

Failure to agree on a suitable list of required capabilities has made it impossible for accreditation and licensing authorities to do their jobs. A lack of demand-side legislation makes any progress on licensing software developers almost inconsequential. Usually, the competence of a software developer is judged by an employer or customer with no help from a licensing authority.

THE NEED FOR CHANGE IS URGENT

The world needs a revolution in software development. The quality of most software produced today is simply terrible. Our phone systems and computer networks are easily compromised. One of the world's richest men had his phone hacked although he is a computer expert. Data that are supposed to be confidential are frequently stolen. Bad software design in the Boeing 737 MAX is blamed for two crashes with hundreds of deaths; a huge number of these new planes were out of service for about two years as a result. Financial and other government system projects frequently fail; some are put into service, with known flaws, long past their due date. Although we hear of software issues every week, many more are not reported to the public.

Many software developers are graduates of educational programs that did not give them the necessary capabilities. On-the-job training often teaches them the bad habits of older developers. Those habits are hard to break.

ACADEMIC FREEDOM AND PROFESSIONAL EDUCATION

One of the most valuable characteristics of our educational system is the right of academics to express ideas without risk of official interference or professional disadvantage. However, many academics have interpreted this "academic freedom" to mean that they can teach whatever they want. Those who hold this opinion sometimes oppose the basic tenet of professional education, namely that an accredited curriculum must give its graduates specified capabilities. Basing professional accreditation on capabilities is a compromise. It allows academics to choose which facts, models, and methods they teach and how they teach those methods, provided that the graduates will have the required capabilities.

LICENSING AND CIVIL LIBERTIES

Some believe that requiring those who want to practice a specific trade or profession to be licensed is a violation of their civil rights. They may even give an example, such as, "If I want to rewire a house for someone else, it is my right to do so." In fact, in many jurisdictions it is not legal to do so unless you are a

licensed electrician. Our lawmakers, in their wisdom, have recognized that improper wiring can cause fires, expose the owners to risk of shock, and even damage power supplies. Requiring that people working as electricians be licensed does not violate our civil rights; it protects us. The same would be true if software developers required a license.

WHAT MUST BE DONE?

Changing software development into an engineering profession requires

- › an agreed list of required capabilities such as that proposed in the article by Landwehr et al.³
- › legally binding demand-side legislation
- › a licensing authority with the legal power to enforce the demand-side legislation, license qualified developers, and discipline developers who do not practice properly
- › an accreditation authority that reviews proposed professional software engineering programs to make sure that the graduates have the capabilities required for practicing the profession.

CHANGE WILL BE STRONGLY OPPOSED BECAUSE MANY PEOPLE ARE DOING VERY WELL WITH "BUSINESS AS USUAL" AND WILL RESENT BEING TOLD THAT THEY NEED TO BE REEDUCATED.

Those who want to establish capability-based licensing of software engineers must be prepared to face both inertia and bitter opposition.

The path to higher quality software requires improving the education and training of almost every software developer. Change will be strongly opposed because many people are doing very well with "business as usual" and will resent being told that they need to be reeducated. They will point to new requirements and techniques and proclaim, "Nobody does it that way."

Most of those who are teaching computer science today were educated in programs that were not designed to prepare students for licensing. Many will not see the need for the changes and will resent any move that might prevent them from teaching their favorite topics.

Some employers will not like the fact that licensed professional engineers are supposed to put public safety before employer profit. It is far easier to manage people who believe that their job is solely to please their employers.

However, our society is far too dependent on software being trustworthy to allow the present "Wild West" of software development to continue. 🤖

REFERENCES

1. P. Naur and B. Randell, Eds., *Software Engineering: Report on a Conference Sponsored by the NATO Science Committee*, Garmisch, Germany, 7–11 October, 1968, Scientific Affairs Division, NATO, Brussels. New York: ACM, Jan. 1969.
2. D. L. Parnas, "Structured programming: A minor part of software engineering," in *Proc. Int. Workshop at the European Joint Conf. Theory Pract. Softw. (ETAPS'03)*, Apr. 6, 2003, pp. 19–25.
3. C. Landwehr et al., "Software systems engineering programmes: A capability approach," *J. Syst. Softw.*, vol. 125, pp. 354–364, Mar. 2017. doi: 10.1016/j.jss.2016.12.016.

DAVID LORGE PARNAS is a professor emeritus at McMaster University, Hamilton, Ontario, Canada, and the University of Limerick as well as president of Middle Road Software, Inc., Ottawa, Ontario, K1V 1V5, Canada. He is a Fellow of IEEE and ACM. Contact him at parnas@mcmaster.ca.

ADVERTISER INFORMATION

Advertising Coordinator

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

Advertising Sales Contacts

Mid-Atlantic US:
Dawn Scoda
Email: dscoda@computer.org
Phone: +1 732-772-0160
Cell: +1 732-685-6068 | Fax: +1 732-772-0164

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

Northeast, Europe, the Middle East and Africa:
David Schissler
Email: d.schissler@computer.org
Phone: +1 508-394-4026

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

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

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

Heather Bounadies
Email: hbonadies@computer.org
Phone: +1 623-233-6575

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

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



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.

JULY

1 July

- ICALT (IEEE Int'l Conf. on Advanced Learning Technologies), Bucharest, Romania

6 July

- ISVLSI (IEEE Computer Society Symposium on VLSI), Nicosia, Cyprus

10 July

- ICDCS (IEEE Int'l Conf. on Distributed Computing Systems), Bologna, Italy

11 July

- ICME (IEEE Int'l Conf. on Multimedia and Expo), Taipei, Taiwan

21 July

- CBMS (IEEE Int'l Symposium on Computer-Based Medical Systems), Shenzhen, China

AUGUST

1 August

- ICCP (Int'l Conf. on Computational Photography), Pasadena, USA

2 August

- MIPR (IEEE Int'l Conf. on Multimedia Information Processing and Retrieval), virtual

4 August

- BCD (IEEE/ACIS Int'l Conf. on Big Data, Cloud Computing,

and Data Science Eng.), Danang, Vietnam

7 August

- CSF (IEEE Computer Security Foundations Symposium), Haifa, Israel

9 August

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

15 August

- RE (IEEE Int'l Requirements Eng. Conf.), Melbourne, Australia

SEPTEMBER

6 September

- CLUSTER (IEEE Int'l Conf. on Cluster Computing), Heidelberg, Germany

12 September

- ARITH (IEEE Symposium on Computer Arithmetic), virtual

18 September

- QCE (IEEE Quantum Week), Broomfield, Colorado, USA

19 September

- AI4I (Int'l Conf. on Artificial Intelligence for Industries), Laguna Hills, USA
- AIKE (IEEE Int'l Conf. on Artificial Intelligence and Knowledge Eng.), Laguna Hills, USA

- ESEM (ACM/IEEE Int'l Symposium on Empirical Software Eng. and Measurement), Helsinki, Finland

- TransAI (Int'l Conf. on Transdisciplinary AI), Laguna Hills, USA

26 September

- ASE (IEEE/ACM Int'l Conf. on Automated Software Eng.), Ann Arbor, USA
- LCN (IEEE Conf. on Local Computer Networks), Edmonton, Canada

OCTOBER

3 October

- ICSME (IEEE Int'l Conf. on Software Maintenance and Evolution), Limassol, Cyprus
- NAS (IEEE Int'l Conf. on Networking, Architecture and Storage), Philadelphia, USA

4 October

- IMET (Int'l Conf. on Interactive Media, Smart Systems and Emerging Technologies), Limassol, Cyprus

16 October

- MODELS (ACM/IEEE Int'l Conf. on Model Driven Eng. Languages and Systems), Montreal, Canada
- VIS (IEEE Visualization and



Visual Analytics), Oklahoma City, USA

NOVEMBER

2 November

- SBAC-PAD (IEEE Int'l Symposium on Computer Architecture and High-Performance Computing), Bordeaux, France

6 November

- IISWC (IEEE Int'l Symposium on Workload Characterization), Austin, USA

7 November

- BIBE (IEEE Int'l Conf. on Bioinformatics and Bioengineering), Taichung, Taiwan

10 November

- ASONAM (IEEE/ACM Int'l Conf. on Advances in Social Networks Analysis and Mining), virtual

11 November

- IPCCC (IEEE Int'l Performance, Computing, and Communications Conf.), Austin, USA

13 November

- SC22 (Int'l Conf. for High-Performance Computing, Networking, Storage and Analysis), Dallas, USA

14 November

- SuperCheck (IEEE/ACM Int'l Symposium on Checkpointing for Supercomputing), Dallas, USA

17 November

- CHASE (IEEE/ACM Conf. on Connected Health: Applications, Systems and Engineering Technologies), Washington, DC, USA

21 November

- ATS (IEEE Asian Test Symposium), Taichung, Taiwan

23 November

- ITNAC (Int'l Telecommunication Networks and Applications Conf.), Wellington, New Zealand

28 November

- ICA (IEEE Int'l Conf. on Agents), Adelaide, Australia
- PRDC (IEEE Pacific Rim Int'l Symposium on Dependable Computing), virtual

29 November

- AVS (IEEE Int'l Conf. on Advanced Video and Signal-Based Surveillance), Madrid, Spain

30 November

- ICDM (IEEE Int'l Conf. on Data Mining), Orlando, USA
- ICKG (IEEE Int'l Conf. on Knowledge Graph), virtual

DECEMBER

5 December

- BigMM (IEEE Int'l Conf. on Multimedia Big Data), Naples, Italy
- ISM (IEEE Int'l Symposium on Multimedia), Naples, Italy
- RTSS (IEEE Real-Time Systems Symposium), Houston, USA
- SMDS (IEEE Int'l Conf. on Smart Data Services), Barcelona, Spain

6 December

- BIBM (IEEE Int'l Conf. on Bioinformatics and Biomedicine), Las Vegas, USA
- UCC (IEEE/ACM Int'l Conf. on Utility and Cloud Computing), Portland, Oregon, USA

7 December

- SNPD (IEEE/ACIS Int'l Winter Conf. on Software Eng., Artificial Intelligence, Networking and Parallel/Distributed Computing), Taichung, Taiwan

13 December

- CloudCom (IEEE Int'l Conf. on Cloud Computing Technology and Science), Bangkok, Thailand

17 December

- Big Data (IEEE Int'l Conf. on Big Data), Osaka, Japan



**Learn more
about IEEE
Computer Society
conferences**

computer.org/conferences

Evolving Career Opportunities Need Your Skills

Explore new options—upload your resume today

www.computer.org/jobs

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

Take advantage of these special resources for job seekers:



JOB ALERTS



TEMPLATES



WEBINARS



CAREER
ADVICE



RESUMES VIEWED
BY TOP EMPLOYERS

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



IEEE
COMPUTER
SOCIETY



IEEE