COMPUTING edge

- Software
- Blockchain
- Augmented and Virtual Reality
- Cyber-Physical Systems

SEPTEMBER 2019

www.computer.org

IEEE

IEEE COMPUTER SOCIETY

COMPUTING edge

IEEE COMPUTER SOCIETY computer.org • +1 714 821 8380

Printed with inks containing soy and/or vegetable oils

IEEE

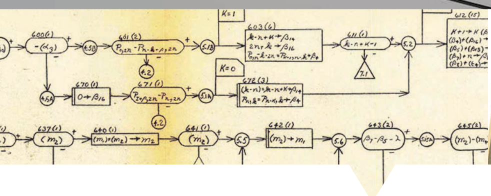
IEEE COMPUTER SOCIETY

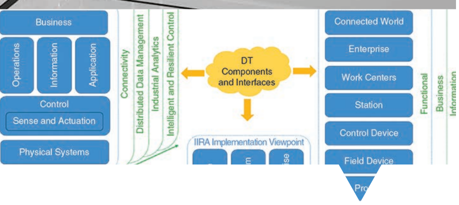
Circulation: ComputingEdge (ISSN 2469-7087) is published monthly by the IEEE Computer Society. IEEE Headquarters, Three Park Avenue, 17th Floor, New York, NY 10016-5997; IEEE Computer Society Publications Office, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720; voice +1 714 821 8380; fax +1 714 821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

SEPTEMBER 2019 • VOLUME 5, NUMBER 9

COMPUTING edge





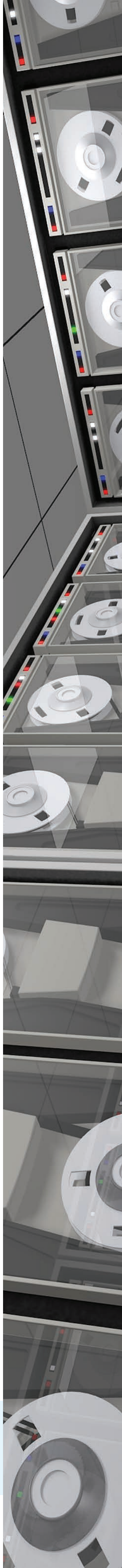
Software

Blockchain

Augmented and Virtual Reality

Cyber-Physical Systems

Departments



Magazine Roundup

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

Computer

Human Eye Movements Reveal Video Frame Importance

Human eye movements indicate important spatial information in static images and videos. However, videos contain additional temporal information and convey a storyline. The authors of this article from the May 2019 issue of *Computer* explore whether eye movement patterns reflect frame importance during video viewing and facilitate video summarization.

Computing in Science & Engineering

Simulation and Experimental Study on the Active Stability of High-Speed Trains

In this article from the May/June 2019 issue of *Computing in Science & Engineering*, an active hunting stability scheme is proposed for high-speed trains based on frame lateral vibration control. The stability of the vehicle is improved by exerting active control force on the front and rear frames of the bogie. First, a simplified lateral vibration model of a single bogie is established to the control system design. The feedback gain matrix is obtained according to the linear quadratic optimal control theory. Then, the multibody dynamics model of the vehicle is built using SIMPACK, and the linear stability and straight running performance are analyzed under different working configurations. Finally, the active control effects are verified by a scaled roller rig.

IEEE Annals of the History of Computing

"The Official Response Is Never Enough"

The Rockefeller Foundation

shipped two Apple II computers with VisiCalc to the Tunisian Ministry of Agriculture to address a grain shortage in the early 1980s. The foundation believed that VisiCalc would enable the speedy and complex analytical modeling necessary to improve the management and, consequently, the production of grain resources. The foundation also argued that VisiCalc would empower individuals in the Ministry of Agriculture, improving their own analytical thinking as they became more familiar with the modeling capabilities of the software. Even with the use of VisiCalc, Tunisia experienced violent riots due to high bread prices after the government's removal of grain subsidies. This article from the January–March 2019 issue of *IEEE Annals of the History of Computing* explores the narratives and uses of VisiCalc in the Tunisian Ministry of Agriculture in addressing this food crisis both before and after the riots.

IEEE Computer Graphics and Applications

Comfortable Immersive Analytics with the VirtualDesk Metaphor

The VirtualDesk metaphor is an opportunity for more comfortable and efficient immersive data exploration, using tangible interaction with the analyst's physical work desk and embodied manipulation of mid-air data representations. In this article from the May/June 2019 issue of *IEEE Computer Graphics and Applications*, the authors present an extended discussion of its underlying concepts, and review and compare two previous case studies where promising results were obtained in terms of user comfort, engagement, and usability. They also discuss findings of a novel study conducted with geovisualization experts, pointing to directions for improvement and future research.

IEEE Intelligent Systems

Robust Authentication Using Dorsal Hand Vein Images

This article from the March/April 2019 issue of *IEEE Intelligent Systems* presents a robust dorsal hand vein authentication system. A new method is proposed for the region of interest extraction using fingertips and finger valley key points. Some new features and a new classifier are proposed based on information set theory. Information set stems from a fuzzy set on representing the uncertainty in its attribute/information source values using the information-theoretic entropy function. The new feature types include vein effective information, vein energy feature, vein sigmoid feature, Shannon transform feature, and composite transform feature. A classifier called the improved Hanman classifier is formulated from training and test feature vectors using Frank t-norm and the entropy function. The performance and robustness are evaluated on GPDS and BOSPHORUS palm dorsal vein databases under both the constrained and unconstrained conditions.

IEEE Internet Computing

Bots Acting Like Humans: Understanding and Preventing Harm

Bots are algorithmically driven entities that act like humans in conversations via Twitter, Facebook, chats, or Q&A sites. This article from the March/April 2019 issue of *IEEE Internet Computing* studies how they can affect online conversations, provides a taxonomy of harm that can be caused, and discusses how to prevent harm by studying when abuses occur.

IEEE Micro

A Hardware Accelerator for Tracing Garbage Collection

Many workloads are written in garbage-collected languages and GC consumes a significant fraction of resources for these workloads. The authors of this article from the May/June 2019 issue of *IEEE Micro* propose decreasing this overhead by moving GC into a small hardware accelerator that is located close to the memory controller and performs GC more efficiently than a CPU. They first show a general design of such a GC accelerator and describe how it can be integrated into both stop-the-world and pause-free garbage collectors. They then demonstrate an end-to-end RTL prototype, integrated into a RocketChip RISC-V System-on-Chip (SoC) executing full Java benchmarks within JikesRVM running under Linux on FPGAs. The prototype performs the mark phase of a tracing GC at 4.2× the performance of an in-order CPU, at just 18.5 percent of the area. By prototyping the design in a real system, they show that the accelerator can be adopted without invasive changes to the SoC, and they estimate its performance, area, and energy.

IEEE MultiMedia

Multipoint Cooperative Transmission for Virtual Reality in 5G New Radio

To meet the requirements of lower latency and massive data transmission in virtual reality (VR) applications, a multipoint cooperative transmission mechanism is proposed for VR applications over 5G New Radio. In particular, different-quality coding levels are utilized through the multipoint cooperative transmission to support the immersive experience of users' different views. Read more in the January–March 2019 issue of *IEEE MultiMedia*.

IEEE Pervasive Computing

Area Occupancy Counting through Sparse Structural Vibration Sensing

This article from the January–March 2019 issue of *IEEE Pervasive Computing* presents an indoor area occupancy counting system utilizing the ambient structural vibration induced by pedestrian footsteps. The system achieves 99.55-percent accuracy in pedestrian footsteps detection, 0.2 people mean estimation error in pedestrian traffic estimation, and 0.2 area occupant activity estimation error in real-world uncontrolled experiments.

IEEE Security & Privacy

Buddy's Wearable Is Not Your Buddy: Privacy Implications of Pet Wearables

As an increasingly prevalent class of consumer device, pet wearables hold more privacy implications than might be initially apparent. Through analysis of privacy policies, the authors of this article from the May/June 2019 issue of *IEEE Security & Privacy* show that more data is captured about owners than pets—and which data is captured remains vague.

IEEE Software

Ethics Is a Software Design Concern

The IEEE and Association for Computing Machinery (ACM) joint report "Software Engineering Code of Ethics" summarizes the responsibilities of software engineers as the following: Software engineers shall commit themselves to making the analysis, specification, design, development, testing, and maintenance of software a beneficial and respected profession. Read more in the May/June 2019 issue of *IEEE Software*.

IT Professional

Cloud-Based Architecture to Implement Electronic Health Record (EHR) System in Pakistan

Electronic health record (EHR) systems are being used in several developed countries to minimize the problems and limitations of the conventional paper-based approach. However, several developing countries like Pakistan have not advanced significantly in adopting the new healthcare standards due to socioeconomic and technological constraints. Although there are some healthcare providers that are using their own EHRs, there is no electronic repository of patients' electronic health data maintained at the government level. In this article from the May/June 2019 issue of *IT Professional*, a cloud-based architecture for the implementation of EHR for hospitals in Pakistan is proposed. Adopting the proposed architecture will help improve patient care, diagnostics, disease presentation, and round-the-clock availability of electronic health information. The development of such a system will not only enable doctors and hospitals to exchange patient information but will also establish an electronic health data repository that subsequently can be used for diverse purposes, such as predictive diagnostics and personalized medicine.

EDITOR'S NOTE

Software's Evolution

We've come a long way from programming mechanical computers using punched cards. More than a century of developments in theory, hardware, programming languages, and methodologies has led to a present in which software is integral to our daily lives. This issue of *ComputingEdge* looks back on the people, programs, and processes that carried software engineering to where it is today.

The author of *IEEE Software*'s "The History of Software Engineering" details the evolution of software engineering from its origins in the 19th century to its modern form—and looks ahead to its future. "Flowcharting Templates," from *IEEE Annals of the History of Computing*, examines a tool that early programmers used to draw diagrams containing inputs, outputs, operations, decisions, and connectors.

Software is the basis of many of today's cutting-edge technologies, including blockchain. *IT Professional*'s "Blockchain and the Economics of Food Safety" describes both the positive impact blockchain could have on food supply chains and the challenges involved in hiring blockchain developers. *IEEE Security & Privacy*'s "Silver Bullet Talks with Nick Weaver" delves into security problems with blockchain technologies, including bugs in code.

Augmented-reality (AR) and virtual-reality (VR) systems are improving in part through software innovations. *IEEE Pervasive Computing*'s "Co-creation and Risk-Taking—In Pursuit of New Technology for Human Augmentation: An Interview with Pranav Mistry" discusses how software advances have enabled various new AR and VR products. *IEEE Computer Graphics and Applications*' "Compressing VR: Fitting Large Virtual Environments within Limited Physical Space" focuses on how to give users larger VR environments.

Software is also integral to cyber-physical systems (CPS). Two *Computer* articles on CPS conclude this *ComputingEdge* issue. "Computer Security as Civil Defense" calls for technology and policy changes to protect against cyber-physical attacks. "Cyber-Physical Systems and Digital Twins in the Industrial Internet of Things" introduces the concept of a digital twin, a virtual representation that serves as the real-time digital counterpart of a physical object or process and addresses every instance for its total lifecycle.

The History of Software Engineering

Grady Booch

THE FIRST COMPUTERS were human (and for the most part, women). The term "digital" didn't enter circulation until around 1942, when George Stibitz took the ideas from another George (Boole) and applied them to electromechanical devices. It took another decade for John Tukey to popularize the term "software." What, then, of the term "software engineering"?

The Origins of the Term

Many suggest it came from the 1968 NATO Conference on Software Engineering, coined by Friedrich Bauer. Others have pointed to the 1966 letter by Anthony Oettinger in *Communications of the ACM*, wherein he used the term "software engineering" to make the distinction between computer science and the building of software-intensive systems.[1] Even earlier, in the June 1965 issue of *Computers and Automation*, there appeared a classified ad seeking a "systems software engineer."

All the data I have points to Margaret Hamilton as the person who first coined the term. Having worked on the SAGE (Semi-automatic Ground Environment) program, she became the lead developer for *Skylab* and *Apollo* while working at the Draper Lab. According to an (unpublished) oral history, she began to use the term "software engineering" sometime in 1963 or 1964 to distinguish her work from the hardware engineering taking place in the nascent US space program.

Software Engineering versus Computer Science

Grace Hopper suggested that programming is a practical art; Edsger Dijkstra called the art of programming the art of organizing complexity; Donald Knuth referred to programming as art because it produced objects of beauty. I suspect that all of these observations are true, but what I like best is David Parnas's observation—much like Anthony Oettinger's—that there is a distinction between "computer science" and the other stuff that we do. This is not unlike the distinction between chemical engineering and chemistry: both are valid; both have their particular sets of practices; both are very different things. Software engineering is, in my experience, equally an art and a science: it is the art of the practical.

Engineering in all fields is all about the resolution of forces. In civil engineering, one must consider static and dynamic forces of a physical nature and of human nature. In software engineering, one also must balance cost, schedule, complexity, functionality, performance, reliability, and security, as well as legal and ethical forces. Computing technology has certainly changed since the time of Charles Babbage. However, the fundamentals of engineering hold true, although, as we shall see, each age discovers some new truth about engineering software.

From the 19th to the 20th Century: Human Computers

Ada Lovelace was perhaps the first person to understand that programming was a thing unto itself. Around that same time, George Boole brought a new way of thinking to the mathematicians and philosophers of the world, as expressed in his classic book *The Laws of Thought*.[2] At the end of the 19th century, we saw the first human computers, such as Annie Cannon, Henrietta Leavitt, and others, the so-called "Harvard Computers" working for the astronomer Edward Pickering. The way these women organized their work was astonishingly similar to contemporary agile development practices; they too had a different way of thinking, very different for their time.

Around the start of the new century, as computational problems began to scale up and as mechanical aids to calculation became more reliable and economical, the process of computing underwent further regimentation. It was common to see large rooms filled with human computers (again, mostly women), all lined up in rows. Data would enter

 2469-7087/19

one end; a computer would carry out one operation and then pass the result to the next computer. This was in effect the organic manifestation of what today we'd call a pipeline architecture.

From the Great Depression to World War II: Birth of the Electronic Computer

Efficiency and the reduction of costs were then, as they are now, important to every industrial process. So, we saw people such as Frederick Taylor and Frank and Lillian Gilbreth (of *Cheaper by the Dozen*[3] fame) introduce time and motion studies. The Gilbreths also promoted the concept of process charts—the direct predecessor of flowcharts—to codify industrial processes. It did not take long for these same ideas in manufacturing to jump over to the problems of computing.

As the global Great Depression took hold, the Works Progress Administration was launched as part of President Roosevelt's New Deal. Gertrude Blanche was put in charge of the Mathematical Tables Project, the predecessor of today's *Handbook of Mathematical Functions*. This was a work relief project that employed hundreds of out-of-work mathematicians and computers (again, mostly women). Blanche's work developed best practices for human computing that were extremely sophisticated, including mechanisms for error checking, which influenced the way early punched-card computing evolved. In 1940, Wallace Eckert published *Punched Card Methods in Scientific Computing*,[4] which turned out to be, in a manner of speaking, the first computing methodology or pattern language.

As the winds of war were gathering in Europe, George Stibitz applied George Boole's ideas of binary logic to build the first digital adder made of electromechanical relays. He called this the K Model (the K representing the kitchen table on which he built it), and thus digital computing was born. The idea of building electromechanical mechanisms for computation spread rapidly, and it was not long thereafter that others realized that relays could be replaced by vacuum tubes, which were much, much faster. In the summer of 1944, a serendipitous meeting between John von Neumann (who at the time was working on the Manhattan Project) and Herman Goldstine (who was working at the Ballistic Research Laboratory) led to their connection with John Mauchly (a professor at the Moore School of Electrical Engineering). This caused ENIAC (Electronic Numerical Integrator and Computer) to come into prominence and, more important, later yielded the *First Draft of a Report on the EDVAC* (Electronic Discrete Variable Automatic Computer).[5]

And thus was born a new way of thinking: the concept of a programmable, electronic computer with its instructions stored in memory.

Grace Hopper, very much in the spirit of Ada Lovelace, then rediscovered the idea that software could be a thing unto itself, distinct from a machine's hardware. This led to one of the first instances of abstraction in programming, the idea that one could devise a programming language at a level closer to human expression and further from the machine's hardware. Furthermore, as Hopper realized, one could use the computer itself to translate those higher-order expressions into machine language; the compiler was born.

> Ada Lovelace was perhaps the first person to understand that programming was a thing unto itself.

In the lamentations of World War II, the computing world split into three pieces. In Germany, there was Konrad Zuse. In a different time and place, his work would have been the center of gravity of modern computing, for he invented the first high-order programming language as well as the first general-purpose stored computer.

In England, there was Bletchley Park, where Alan Turing laid the theoretical foundations for modern computer science. However, it took an engineer—most notably Tommy Flowers—to turn those theories into pragmatic solutions, and from this Colossus was born. Dorothy Du Boisson, a human computer, served as the primary operator of Colossus. In her experience of leading a team of women who operated Colossus, she codified the ideas of workflow that eventually were programmed into the machine itself.

In the US, ENIAC, then later EDVAC, dominated the scene. Initially, "programming" was carried out by wiring up plugboards, a task carried out by human computers (yet again, mostly women), such as Kay Antonelli, Betty Snyder, Frances Spence, Ruth Teitelbaum, and Marlyn Wescoff. The way they organized their work was reminiscent of the Harvard Computers and thus, in a manner of speaking, anticipated the structure of contemporary small development teams focused on continuous integration.

Post World War II: Rise of Computing and Birth of Software Engineering

The technical and economic forces that would shape modern software engineering further coalesced in the economic rise at the end of World War II, where we began to see computing applied to problem domains beyond the needs of conflict. Herman Goldstine built on the ideas of the Gilbreths and, together with John von Neumann, invented a notation that eventually morphed into what today we call flowcharts. Maurice Wilkes, David Wheeler, and Stanley Gill invented the concept of subroutines, thus again raising computing's levels of abstraction, and making manifest the pragmatics of algorithmic decomposition. John Backus took Grace Hopper's early work and went further, yielding Fortran, the high-level imperative language that would dominate scientific computing for years to come.

The commercial world, now unleashed at the end of global conflict, turned to automatic aids to computing: opportunities for growth quickly outran the cost and reliability of human computers. The first computer put in commercial use was the Lyons Electronic Office (LEO). John Pinkerton, LEO's chief engineer, had the insight that software could be treated as a component unto itself. Realizing that many low-level programming tasks kept being written over and over again, he began to bundle these common routines into libraries, forming what today we'd call an operating system or framework, yet another rise in programming's levels of abstraction.

Grace Hopper, Robert Bemer, Jean Sammet, and others, influenced by John Backus's work, created Cobol, another imperative language, focused on the needs of businesses. With the introduction of IBM's System/360, it was now possible to write software for more than one specific machine. IBM's decision to unbundle software from hardware was a transformative event: now it was possible to develop software as a component that had individual economic value. Around this time, organizations such as SHARE emerged—a predecessor of today's open source software movement—giving a platform for third parties to write software for hardware they themselves didn't control. In the UK, Dina St. Johnson seized on the business opportunity and established England's first software services business. This made manifest the idea that one could outsource software development to teams with particular computing skills a company with specific domain knowledge might not possess.

Rise of the Cold War: Coming of Age

The rise of the Cold War between the US and the Soviet Union generated another set of forces that pushed software engineering to come of age. Tom Kilburn and his work with Whirlwind explored the possibilities of real-time programming, and that work led directly to the SAGE system. Constructed as a defense against the Soviet threat of sending nuclear-armed bombers over the Arctic, SAGE led to a number of important innovations and issues, including

- human–computer interfaces using CRT displays and light pens,
- the institutionalization of core memory, and
- the problems associated with building very large software systems in a distributed environment.

Software development was no longer just a small part of bringing a computer to life; it was increasingly a very expensive part, and certainly the most important part.

So there we were, in the second half of the 1960s, with the confluence of three important events in the history of software:

- the rise of commercial software as a product unto itself,
- the complexities of defense systems such as SAGE, and
- the rise of human-critical software as demanded by the US space program.

This is the context in which Margaret Hamilton coined the term "software engineering" and in which NATO declared that there was a "software crisis."

A sort of programming priesthood was the common form of software development at the time, and—in its time—it made a great deal of sense. In that era, the cost of a computer was greater than the cost of its programmers, and as such, computers would be kept apart in a climate-controlled room. Much like the pipelined methods of the punched-card era, analysts would take requirements and pass them on to programmers, who would use their flowcharts to devise algorithms. These programmers in turn would pass their programs on to keypunchers. The resulting card decks would be given to the computer operators working in their sacred space.

It wasn't until the economics of computers changed with the rise of minicomputers and microcomputers, together with the realization of Christopher Strachey's idea of time sharing, that this model of development changed. This is also the context in which the basic principles of software project management came alive, as Fred Brooks so profoundly described in *The Mythical Man Month*.[6] Brooks made the important insight that software engineering was not just a technical process but also a very human process.

The economic rise after World War II, given a further boost by the Cold War, led inevitably to a counterculture shift, as wonderfully described by John Markoff in *What the Dormouse Said*.[7] The introduction of personal computing not only was fueled by technical and social advances but also changed the nature of software engineering. Now, programmers were more expensive than computers, and it was economically viable to put computers everywhere. This led to Allen Newell speaking of the enchanted world that computing made possible, as described in his wonderful essay "Fairytales."[8]

From the Sixties to the Eighties: Maturation

Software engineering was forced to mature. Larry Constantine was perhaps the first to introduce the concept of modular programming, with the ideas of coupling and cohesion applied as a mechanism for algorithmic decomposition. Edsger Dijkstra took a more formal approach, giving us an important tool for software engineering: the idea of structured programming.

Around the same time, there was important work by researchers such as Robert Floyd and Tony Hoare, who devised formal ways to express and reason about programs—a true attempt to connect computer science and software engineering. Niklaus Wirth invented Pascal, an effort to explicitly support best practices in structured programming. Ole Dahl and Kristen Nygaard had the outrageously wonderful idea that yielded the invention of Simula, a language that was object-oriented rather than algorithmic in nature.

Winston Royce then brought to us the idea of a formal software development process. Although he is much criticized for what we today call the waterfall process, his methodology was actually quite advanced: he spoke of iterative development, the importance of prototyping, and the value of artifacts beyond source code itself. Coupled with David Parnas's ideas of information hiding, Barbara Liskov's ideas of abstract data types, and Peter Chen's approaches to entity–relationship modeling, all of a sudden the field had a vibrant set of ideas whereby to expresses the artifacts and the processes of software development.

This led to the first generation of software engineering methodologies. Doug Ross, Larry Constantine, Ed Yourdon, Tom DeMarco, Chris Gane, Trish Sarson, and Michael Jackson—to name just a few—developed methods for structured analysis and design that took over the field. Adding the work by Michael Fagan (on software inspections), James Martin (on information engineering), John Backus (on functional programming), and Leslie Lamport (on best practices for distributed computing), software engineering entered in its first golden age.

> The introduction of personal computing changed software engineering.

The Eighties and Onward: Golden Age

However, a sea change was coming. Owing to the growing problems of software quality, the rise of ultralarge software-intensive systems, the globalization of software, and the shift from programs to distributed systems, new approaches were needed. Ole Dahl and Kristen Nygaard's ideas of object-oriented programing gave rise to a completely new class of programming languages: Smalltalk, C with Classes, Ada, and many others. Although structured methods were useful, they were not altogether sufficient for these new languages, and thus was born the second golden age of software engineering.

Ada—the US Department of Defense's solution to the problem of the proliferation of programming languages and the changing nature of software itself—proved to be a catalyst for this era. Some of the structured-method pioneers pivoted. James Martin and Ed Yourdon celebrated object-oriented approaches; others brought completely new ideas to the field: Stephen Mellor, Peter Coad, and Rebecca Wirfs-Brock, to name a few. The Booch Method grew out of this primordial soup of ideas, as did Jim Rumbaugh's OMT (object-modeling technique) and Ivar Jacobson's Objectory. Sensing an opportunity to bring the market to some common best practices, the three of us united to produce what became the Unified Modeling Language (made an Object Management Group standard in 1987) and then the Unified Process.

Other aspects of software engineering came into play—for example,

- Philippe Kruchten's 4+1 View Model of software architecture;
- Barry Boehm's work in software economics, together with his spiral model;
- Vic Basili and his ideas on empirical software engineering;
- Capers Jones and software metrics;
- Harlan Mills and clean-room software engineering;
- Donald Knuth's literate programming; and
- Watts Humphrey and his Capability Maturity Model.

Simultaneously, these software engineering concepts influenced the development of an entirely new generation of programming languages. Bjarne Stroustrup's C with Classes grew up to become C++, which later influenced the creation of Java. Alan Cooper's Visual Basic invigorated the Windows platform. Brad Cox's invention of Objective-C had a tremendous effect on NeXT and Apple. Furthermore, Cox's ideas surrounding component-based engineering—another rise in software engineering's levels of abstraction—led directly to Microsoft's OLE (object linking and embedding) and COM (Component Object Model), which were the predecessors of today's microservice architecture.

The Nineties and the Millennium: Era of Disruptions

But another change was in the wind: the Internet. Suddenly we had a very rich, as of yet unexplored, platform. In it, distribution was the default, consumers were the new stakeholders, users were measured in the billions, and participants in this ecosystem were not necessarily reliable or trustworthy. We were no longer building programs; we were building systems, often made of parts that we no longer controlled.

By this time, there existed a relatively stable and economically very vibrant software engineering community. Independent companies existed to serve the needs of requirements analysis, design, development, testing, and configuration management. Continuous integration with incremental and iterative development was becoming the norm. The Gang of Four—Eric Gamma, Richard Helm, Ralph Johnson, and John Vlissides—gave us another bump up in software engineering levels of abstraction in the form of the design pattern. Institutionalized by the Hillside Group in 1993, patterns heavily influenced that generation of software development. Jim Coplien took the ideas of software design patterns and applied them to organizational patterns. Mary Shaw and David Garlan furthered these concepts in their work on software architecture styles.

Two other lasting developments of note took place in this era. First, Eric Raymond evolved an important legal framework for open source, making it possible to scale the ideas first seen in the early days of computing, with SHARE. Kiran Karnik, working in India, established the first outsourcing contracts between General Electric and India, thus laying the foundation for a transformative economic shift in software development.

With the Internet well in place and organizations beginning to embrace its possibilities, mobile devices hit the scene, and the world changed yet again. The foundation laid by Brad Cox for component-based engineering morphed into service-based architectures, which in turn morphed into microservice architectures, evolving as the Web's technical infrastructure grew in fits and starts. New programming languages came and went (and still do), but only a handful still dominate—for example, Java, JavaScript, Python, C++, C#, PHP, and Swift. Computing moved from the mainframe to the datacenter to the cloud, but coupled with microservices, the Internet evolved to become the de facto computing platform. Company-specific ecosystems rose like walled cathedrals: Amazon, Google, Microsoft, Facebook, Salesforce, IBM—really, every economically interesting company built its own fortress.

This was now the age of the framework. Long gone were the religious battles over operating systems. Now, battles were fought along the lines of the veritable explosion of open source frameworks: Bootstrap, jQuery, Apache, NodeJS, MongoDB, Brew, Cocoa, Caffe, Flutter—truly a dizzying, ever-growing collection.

Today, we no longer build just programs or monolithic systems; we build apps that live on the edge and interact with these distributed systems. Agile methods—in various personality-led variations—have flowered and have become the dominant method, in name if not necessarily perfectly in practice. Hirotaka Takeuchi and Ikujiro Nonaka coined the term “Scrum” in 1986 as an agile approach to product development. Later, Ken Schwaber and (independently) Jeff Sutherland and Jeff McKenna codified those principles in the domain of software development. Around that same time, Kent Beck introduced the concept of Extreme Programming, while Ralph Johnson further developed the idea of refactoring (which Martin Fowler further codified in his book *Refactoring: Improving the Design of Existing Code*). In February 2001, 17 agilists met in Snowbird, Utah, and penned the Agile Manifesto. The agile approach to software development entered the mainstream.

Software engineering had entered another golden age. Git and GitHub emerged; Joel Spolsky gave us Stack Overflow; Jeannette Wing introduced the idea of computational thinking; Andrew Shafer and Patrick Debois brought us the idea of DevOps; the full stack developer became a thing; the Internet of Things appeared in every imaginable corner of the world. Now, all of a sudden, everyone could learn how to code (and many did).

Artifacts such as SWEBOK (Software Engineering Body of Knowledge, first released in 2004 and whose current version was released in 2014)[9] and the Systems Engineering Body of Knowledge by INCOSE[10] exist as an attempt to codify software engineering best practices.

The Decade Ahead: Big Data and the New Season of AI

But software engineering is about to undergo yet another change.

The foundations of AI have been around for decades. Over the decades, we’ve seen at least four seasons of AI, manifested by the extreme rising and falling of fortunes. What we have now feels different. The growth of big data, the abundance of raw computational power, and the presence of these walled cathedrals have given rise to economic forces that have made first statistical approaches and now neural networks viable. Most of these modern advances have been in what I call “signal AI”: the use of neural networks and gradient descent to do complex pattern matching in images, video, and audio signals. The early outcomes are impressive, as evidenced in IBM’s Watson and Google’s AlphaGo. In many ways, we are just beginning to understand what is possible and where the limits of these connectionist models of computation live.

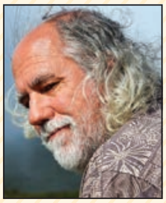
We as an industry have not yet built enough of these AI systems to fully understand how they might impact the software engineering process, as they most certainly will. What is the best lifecycle for systems whose components we teach, rather than program? How do we test them? Where does configuration management fit in when data for ground truth is perhaps more important than the neural network itself? How do we best architect systems with parts whose operation we cannot explain or fully trust?

This will be the challenge of the next generation of women and men who keep software engineering vibrant. Add to this mix the growth of quantum computing, augmented reality, virtual reality, and the spread of computing to every human, every device, and every nook and cranny of the earth and beyond, and this makes for a tremendously exciting time to be in computing.

> How do we best architect systems with parts whose operation we cannot explain or fully trust?

In the history of computing, we have seen the progression of systems from mathematical, to symbolic, to what Yuval Harari calls “imagined realities.” Some software is like building a doghouse: you just do it, without any blueprints, and if you fail, you can always get another dog. Other software is like building a house or a

ABOUT THE AUTHOR



GRADY BOOCH is an IBM Fellow and one of UML's original authors. Contact him at egrady@booch.com and on Twitter as @grady_booch.

high-rise: the economics are different, the scale is different, and the cost of failure is higher. Much of modern software engineering is like renovating a city: there is room for radical innovation, but you are constrained by the past as well as the cultural, social, ethical, and moral context of everyone else in the city.

One thing I do know. No matter the medium or the technology or the domain, the fundamentals of sound software engineering will always apply: craft sound abstractions; maintain a clear separation of concerns; strive for a balanced distribution of responsibilities; seek simplicity. The pendulum will continue to swing—symbolic to connectionist to quantum models of computation; intentional architecture or emergent architecture; edge or cloud computing—but the fundamentals will stand.

I have named a few dozen women and men who have shaped software engineering, but please know that there are thousands more who have made software engineering what it is today, each by his or her own unique contributions. And so it will be for the future of software engineering. As I said in closing in my keynote at the 2015 International Conference on Software Engineering in Florence, software is the invisible writing that whispers the stories of possibility to our hardware.

And you are the storytellers.

Acknowledgments

This essay is based on my ACM Learning Webinar of the same title, broadcast on 25 April 2018. A recording is available at https://www.youtube.com/watch?v=QUz10Z1AfLc. A more extensive bibliography is available as a Web Extra at https://extras.computer.org/extra/mso2018050108s1.pdf.

References

1. A. Oettinger, "President's Letter to the ACM Membership," *Comm. ACM*, vol. 9, no. 8, 1966.
2. G. Boole, *The Laws of Thought*, Walton and Maberly, 1854.
3. F.B. Gilbreth Jr. and E. Gilbreth Carey, *Cheaper by the Dozen*, Thomas Y. Crowell, 1948.
4. W.J. Eckert, *Punched Card Methods in Scientific Computing*, Thomas J. Watson Astronomical Computing Bureau, Columbia Univ., 1940.
5. J. von Neumann, *First Draft of a Report on the EDVAC*, US Army Ordinance Department and Univ. Pennsylvania Moore School of Electrical Eng., 1945.
6. F. Brooks, *The Mythical Man Month: Essays on Software Engineering*, Addison-Wesley, 1975.
7. J. Markoff, *What the Dormouse Said*, Penguin, 2005.
8. A. Newell, "Fairytails," Carnegie Mellon Univ., 1976; http://repository.cmu.edu/cgi/viewcontent.cgi?article=3321&context=compsci.
9. *The Guide to the Software Engineering Body of Knowledge*, IEEE Computer Soc., 2004.
10. *Guide to the Systems Engineering Body of Knowledge (SEBoK)*, Int'l Council Systems Eng., 2012.

This article originally appeared in IEEE Software, *vol. 35, no. 5, 2018.*

Department

Flowcharting Templates

Peggy Aldrich Kidwell
National Museum of American History
Smithsonian Institution

Editor: David Walden, dave.walden.family@gmail.com

The Smithsonian has begun to post online descriptions of objects in its collections. Figures and accounts are now available for roughly 2500 objects from the math and computer collections—perhaps a quarter of the total. Our focus has been on small objects that are easily examined. Hence I recently cataloged the two dozen flowcharting templates in the collections. This brief account is intended to encourage readers to add their own reminiscences to the online records, creating a much richer resource.

Early commercial computer manufacturers, most notably Remington Rand UNIVAC (previously Eckert Mauchly Computer Company, later Remington Rand UNIVAC, now UNISYS) and IBM, faced the challenge of teaching both potential customers and budding programmers about the logical structure of computer programs. Toward that end, they used diagrams called flowcharts, which had separate symbols to represent forms of input and output, operations, decisions, connectors, and directions of logical flow. Thomas Haigh, Mark Priestley, and Crispin Rope, in their book *ENIAC in Action*,[1] have looked specifically at the diagrams used by those programming the pioneering ENIAC computer at the University of Pennsylvania at about the time of World War II. As Nathan Ensmenger has noted in 2016 in *Information and Technology* article on the history flowcharts,[2] Betty Holberton, who worked as a programmer on the ENIAC, took the flowcharting techniques developed at Penn to the first American commercial computer company, Eckert Mauchly. By 1949, EMCC employee Grace Murray Hopper and her colleagues were preparing flow charts for the company's UNIVAC computer under Holberton's direction (see Figure 1). The following year, EMCC copyrighted a set of "Flow Chart Symbols."

The technique proved sufficiently successful for computer companies to introduce and distribute small rectangular plastic templates with the symbols used on flowcharts cut out of them. The company name also featured prominently. The earliest flowcharting template in the Smithsonian collections dates from about 1955 and was distributed, appropriately enough, by Remington Rand Univac, the corporate descendent of EMCC (see Figure 2). The symbols on it are not labeled but are an expanded version of those used in earlier Eckert-Mauchly diagrams. The template is shown in a short, undated, movie entitled *Remington Rand Presents UNIVAC*. It also appeared on the cover of a company publication known as *the Programmer* in March–April, 1956 (see Figure 3).

Other computer manufacturers also soon issued flowcharting. Examples in the Smithsonian collections come from the Electrodata Division of Burroughs, Burroughs itself, IBM, RCA, Honeywell, the Massachusetts firm of Sprague Electric, RCA, Bunker-Ramo Corporation, the Bell System, and Control Data Corporation. Use was not confined

Digital Object Identifier 10.1109/MAHC.2019.2893719
Date of current version 14 March 2019.

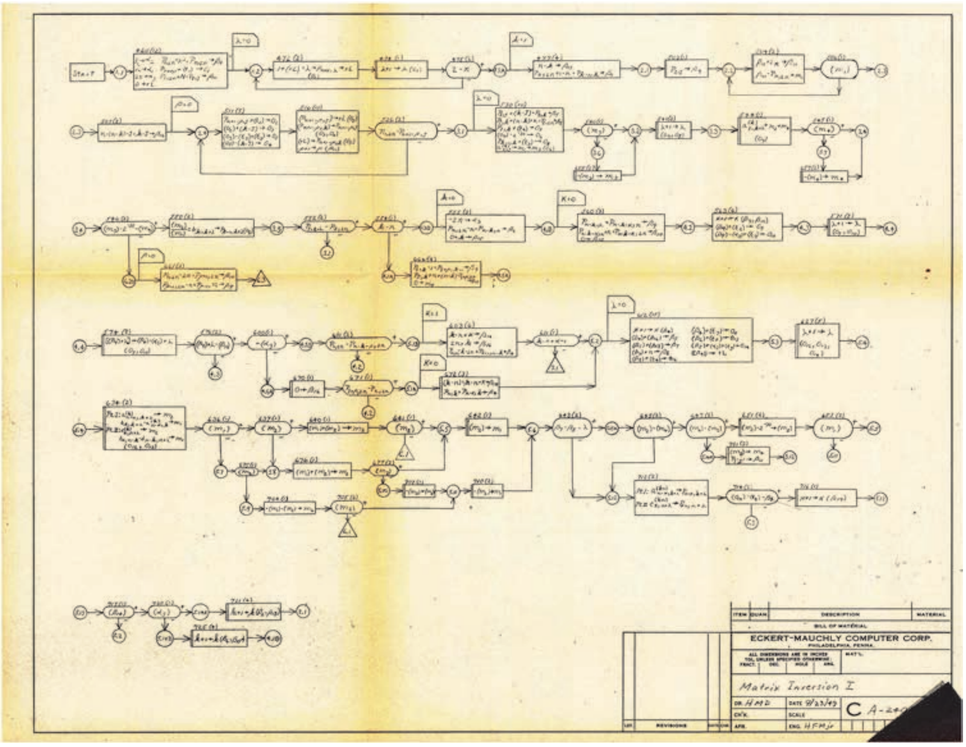


Figure 1. Flowchart drawn by Helen M. Diehl for engineer Herbert F. Mitchell at the Eckert-Mauchly Computer Corporation in September of 1949. Grace Murray Hopper Collection, 1944–1965, Archives Center, National Museum of American History. Smithsonian Image AC0324-0000042.

to American makers of mainframe computers. For example, one IBM flowcharting template was designed for users of the IBM 402 and IBM 403 electronic accounting machines. Somewhat later, the Educational Services affiliate of minicomputer manufacturer Digital Equipment Corporation distributed flowcharting templates. The technique of flowcharting—and the design of templates—also reached internationally to include one distributed by the British firm of ICL.

Once templates had become a standard tool of computer programming, they also were sold

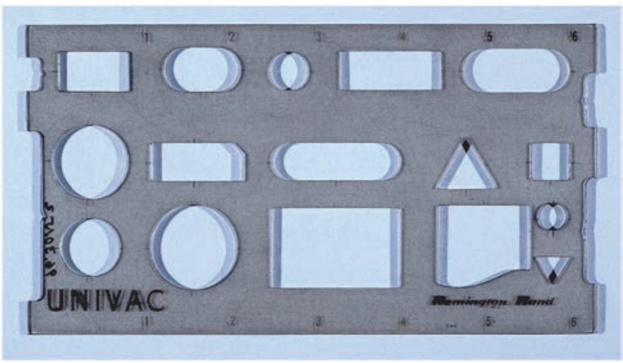


Figure 2 . Remington Rand UNIVAC Flowcharting Template, about 1955, Gift of Joan P. Nichols. Mathematics Collections, National Museum of American History. Smithsonian Image 2003-20240.

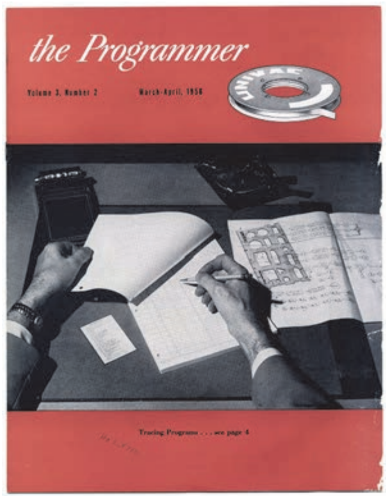


Figure 3. Cover of the Remington Rand publication *The Programmer*, vol. 3 #2, March–April, 1956. Grace Murray Hopper Collection. Smithsonian Image AC0554-000002.

by makers of drawing instruments. Two examples in the Smithsonian collections were made in the U.S., distributed by the German firm of Mars Staedler, and used in Canada. By the 1980s, such templates also might be distributed as give-aways by prospective employers.

This profusion of flow charts led to attempts at standardization. In the early 1960s, the American Standards Association established sectional committee X3 to develop standards for computers and information processing. The first standard developed by the subcommittee on problem description and analysis concerned flowcharting symbols. A proposal circulated in 1963 was approved as ASA Standard X3.5-1965 (the 1965 version of the fifth standard developed by committee X3), and was soon revised as X3.5-1966 and then as X3.5-1970. Templates sometimes refer to the standard used in creating them, offering a clue as to the date on which they were designed.

Surviving flowcharting templates well represent the emergence of commercial computer manufacturers, early attempts to provide training for programmers, and efforts to develop standards within information processing. It is less clear how people actually used them—and surviving objects are mute on the matter! The objects are described online at the museum's website (see http://americanhistory.si.edu/collections/object-groups/flowcharting-templates). These pages have space for comments, and I urge those wishing to share their experiences with flow charts and flowcharting templates to add their comments and memories.

REFERENCES

1. T. Haigh, M. Priestley, and C. Rope, *ENIAC in Action: Making and Remaking the Modern Computer*, Cambridge, MA, USA: MIT Press, 2016.
2. N. Ensmenger, 'The Multiple Meanings of a Flowchart' *Info. Culture: A J. History*, vol. 41, no 10, 2016, pp. 321–351.

Peggy Aldrich Kidwell is Curator of Mathematics at the Smithsonian Institution's National Museum of American History. She has never used a flowcharting template herself, and would welcome comments on those shown on the NMAH website. Contact her at kidwellp@si.edu.

This article originally appeared in
IEEE Annals of the History of Computing, *vol. 41, no. 1, 2019*.

ADVERTISER INFORMATION

Department

Blockchain and the Economics of Food Safety

Nir Kshetri
University of North Carolina at Greensboro

***Abstract*—Blockchain technology has a potential to address many of the food safety challenges facing the world today. Some of the most promising blockchain applications developed to date have been in the food supply chains.**

Adulterated, contaminated, mislabeled, and misbranded food products have imposed tremendous social and economic costs to the global economy. About 600 million people in the world become ill due to contaminated food every year. Of those, about 420 000 die, which include 125 000 children under the age of 5 years.[1] According to a study conducted by the World Bank, unsafe food products cost low- and middle-income economies $110 billion in lost productivity and medical expenses annually (https://www.foodsafetynews.com/2018/10/unsafe-food-in-lmics-costs-110-billion-a-year-world-bank/). One estimate suggested that 30%–40% of the food consumers eat is either "adulterated or mislabeled" (http://www.connect.catalyst-inc.org/techwatch/arcnet). In a survey, 39% of food manufacturers thought that their products can be easily counterfeited, and 40% viewed that food fraud is difficult to detect using the currently available methods (https://www.nzherald.co.nz/business/news/article.cfm?c_id = 3&objectid = 12039985).

Digital Object Identifier 10.1109/MITP.2019.2906761
Date of current version 21 May 2019.

Food safety is of particular concern in industrialized countries where consumers are increasingly demanding higher quality and safer food.[2] A study conducted at the household level in the U.S. found that inadequate quality of food products is one of the key sources of food insecurity.[3]

There is big hope that blockchain technology can address many of the food safety challenges facing the world today.[4] Indeed, some of the most promising blockchain applications outside finance are being developed to address various concerns in the food supply chains (http://internetofthingsagenda.techtarget.com/blog/IoT-Agenda/Blockchain-for-industrial-enterprises-Hype-reality-obstacles-and-outlook).

At the national level, there is a tremendous positive economic impact of safe and quality food products associated with better health outcomes of citizens. At the firm level, companies in the food supply chains can use blockchain to

Table 1. Blockchain deployment in food supply chains: Some examples.

Food retailer	Explanations	Remarks
U.S. retailer Walmart	2016: Trial-tested a blockchain-based solution to monitor pork products in China and produce imported to the U.S. from Latin America (https://classic.qz.com/perfect-company-2/1146289/the-worlds-biggest-retailer-wants-to-bring-blockchains-to-the-food-business).	Blockchain enabled the tracking of pork products in a few minutes compared to many days taken in the past. Details about the farm, factory, batch number, storage temperature, and shipping can be viewed on blockchain (*http://www.foodsafetynews.com/2017/03/a-new-era-of-food-transparency-with-wal-mart-center-in-china/#.WOB65mcVjIU*).
French retailer Carrefour	Signed an agreement with IBM to use the solution.	Announced a plan to track its own branded products in France, Spain, and Brazil. It also noted plans to expand to other countries by 2022 (https://hawthorncaller.com/ibms-food-blockchain-is-going-live-with-a-supermarket-giant-on-board/).
Chinese e-commerce company Jingdong (JD.com)	Implemented blockchain in food supply chains system, mainly involving B2B e-commerce.	2017: Its blockchain system went live with inner Mongolia-based food supplier Kerchin as its first supply-chain partner (Kshetri and Loukojanova, 2019).[6] Kerchin collects and stores data in its supply chain by scanning barcodes of its products. The information is then entered onto blockchain. After that, any changes in data require a digital signature. Both parties are informed if there is any change and modification in the data.[7]
U.S.-based Bumble Bee Foods	March 2019: Announced the launch of a blockchain platform to trace seafood. The company teamed up with German technology company SAP for the project (https://cointelegraph.com/news/north-american-seafood-firm-to-use-blockchain-tech-in-supply-chain).	By scanning a QR code on the product package, consumers would be able to access information related to the details of the supply chain such as products' origins, the size of the catch, the point of capture shipping history, and trade fishing certification.

address problems related to inefficiency, opacity, and fraud. Blockchain is also being used by some firms in the food industry to enhance reputational value by demonstrating their ability to innovate.[5]

Some Blockchain Projects in Food Supply Chains

A number of firms in the food industry have started to incorporate blockchain in supply chains (see Table 1). In November 2018, IBM commercially launched its blockchain-based Food Trust. Companies of all sizes in the food industry supply chain can join the network for a subscription fee that ranges from $100 to $10 000 a month (https://hawthorncaller.com/ibms-food-blockchain-is-going-live-with-a-supermarket-giant-on-board/). IBM Food Trust is being used by many large food companies such as Nestle, Unilever, and Walmart.

The French retailer Carrefour has been one of the early adopters of the IBM Food Trust (see Table 1). The retailer announced in March 2019 that it would launch blockchain-enabled QR-codes for some of its milk products. With a smartphone app, customers can scan the labels to learn details about milk products that they buy. The labels provide relevant details such as the date and location of collection and packaging of a milk package, the GPS coordinates of dairy farm producing it, and how the cow was fed (https://thenextweb.com/hardfork/2019/03/06/carrefour-blockchain-milk/).

Big food retailers are also forcing their suppliers to adopt blockchain. In September 2018, Walmart announced that it would require its suppliers of leafy green vegetables to upload their data to the blockchain system by September 2019 (https://techcrunch.com/2018/09/24/walmart-is-betting-on-the-blockchain-to-improve-food-safety/).

Firms in food supply chains are rapidly adopting blockchain systems. Examples from retailers such as Carrefour indicate that blockchain can be used to provide access to rich and detailed information about food products, which is likely to reduce uncertainty about quality and ingredients. This will increase consumers' confidence in food products that they buy. Food companies, thus, can boost revenue and profits by using blockchain.

For food retailers, another key benefit of blockchain is its ability to effectively handle a crisis situation. To illustrate this argument, consider the 2015 E.coli outbreak at Chipotle Mexican Grill outlets. The crisis left 55 customers ill. There were many negative news stories about this food-borne illness. Many Chipotle restaurants were shut down, and investigations took place. All these led to a significant blow to the reputation of the company. There was a dramatic reduction in sales revenues. The company's share price dropped by 42%. The roots of the problem lie partly in Chipotle's reliance on multiple suppliers. Companies such as Chipotle cannot monitor their suppliers in real time. It is, thus, impossible to prevent food contaminations. It is also difficult to contain a food crisis in a targeted way after it is discovered (*https://hbr.org/2017/03/global-supply-chains-are-about-to-get-better-thanks-to-blockchain*). Chipotle's value proposition is centered on fresh and locally sourced ingredients. Food supply chain systems based on nonblockchain methods are expensive and cumbersome. The process involves manual verification and massive record keeping. Blockchain can reduce the workload and ensure traceability.

Some Key Challenges

While the various benefits of blockchain in food supply chains cannot be disputed, it is also important to look on the cost sides. The high cost of hiring blockchain developers leads to adverse economics of blockchain deployment in this industry. For instance, according to the job data analytics firm Burning Glass Technologies, the median annual salary for a fulltime blockchain developer in the U.S. was $140 000 in 2018, compared to $105 000 for general software developers. Blockchain specialists are reported to charge as much as $250 per hour.[8]

Labor and skill shortages have been identified as a key challenge in the blockchain industry. The shortage is especially severe in developing economies.[9] For instance, out of the country's 2 million software developers, only 5000 were estimated to have blockchain skills. Some speculate that about 80% of these developers may pursue job opportunities outside the country.[10]

Due primarily to the high costs and limited availability of blockchain talents, currently, the deployment of blockchain-based solutions is more justifiable and more realistic in high-value food products than in cheaper products.[4] For instance, in 2018, JD.com announced a plan to implement blockchain to track its meat supply chains. Customers would be able to monitor their meat products. Initial focus would be on high-end beef from Australia (https://tinyurl.com/y8kfyv75). Likewise, the French retailer Carrefour 's traceability project focused on its premium farm products (*https://www.ledgerinsights.com/oxfam-blockchain-cambodian-rice-farmers/*).

A related point is that only big firms in the food supply chains are currently in a position to implement blockchain-based solutions. For instance, JD's SC partner Kerchin that has adopted blockchain had $300 million in revenue in 2017.[7] Most of the food products in developing economies such as Africa and China, on the other hand, are produced by very small farms. These farms lack access to technology or Internet connectivity. Adoption of blockchain systems can be unrealistic for these farms, at least in the near future.

Summary

Blockchain systems can bring transparency and accountability in food supply chains. Such systems, thus, are likely to play a tremendously important role in ensuring food safety. Global economic and health benefits of blockchain systems' deployment to trace food products are, thus, extremely high.

Firms in the food industry can significantly enhance customer loyalty and sales growth by using blockchain. For firms in the food industry, it is also important to be able to handle crisis situations in order to be profitable. Blockchain-based solutions can help deal with risk situations involving crises and emergencies. For instance, if contaminated food products are found, food retailers can easily identify the source and engage in strategic removals of affected products. They do not need to recall the entire product line.

While the benefits of blockchain systems in food supply chains outweigh the costs on average, such systems are currently out of reach for most small firms in the food industry. The solutions already available hold the promise of developing cheaper systems that are easier to use and trust—for farmers, food processing plants, and customers alike. Over time, blockchain implementation costs in the food industry are likely to reduce. This is likely to make blockchain-based solutions more affordable to smaller companies and accelerate its diffusion in food supply chains.

REFERENCES

1. World Health Org., "More than 23 million people in the WHO European region fall ill from unsafe food every year," 2015. [Online]. Available: http://www.euro.who.int/en/health-topics/disease-prevention/food-safety/news/news/2015/12/more-than-23-million-people-in-the-who-european-region-fall-ill-from-unsafe-food-every-year
2. J. Kinsey, "GATT and the economics of food safety," *Food Policy*, vol. 18, no. 2, pp. 163–176, Apr. 1993. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/0306919293900246
3. P. Webb, J. Coates, E. Frongillo, B. Rogers, A. Swindale, and P. Bilinsky, "Measuring household food insecurity: Why it's so important and yet so difficult to do," *J. Nutrition*, vol. 136, no. 5, pp. 1404S–1408S, 2006. [Online]. Available: https://academic.oup.com/jn/article/136/5/1404S/4670058
4. N. Kshetri, "Blockchain systems are tracking food safety and origins," Nov. 21, 2018. [Online]. Available: https://theconversation.com/blockchain-systems-are-tracking-food-safety-and-origins-106491
5. M. Higginson, M. Nadeau, and K. Rajgopal, "Blockchain's Occam problem," Jan. 2019. [Online]. Available: https://www.mckinsey.com/industries/financial-services/our-insights/blockchains-occam-problem?cid=other-eml-alt-mip-mck&hlkid=f1ff7216a70e4041951d60293978a0ea&hctky=2762145&hdpid=95e9bdfa-0709-4b4d-8252-f401bcaac86d
6. N. Kshetri and E. Loukoianova, "Blockchain adoption in supply chain networks in Asia," *IEEE IT Profess.*, vol. 21, no. 1, pp. 11–15, Jan./Feb. 2019.
7. E. Huang, "Blockchain could fix a key problem in China's food industry: The fear of food made in China," 2017. [Online]. Available: https://qz.com/1031861/blockchain-could-fix-a-key-problem-in-chinas-food-industry-the-fear-of-food-made-in-china/
8. L. Mearian, "Blockchain moves into top spot for hottest job skills," May 1, 2018. [Online]. Available: https://www.computerworld.com/article/3235972/blockchain-moves-into-top-spot-for-hottest-job-skills.html
9. N. Kshetri, "The Indian blockchain landscape: Regulations and policy measures," *Asian Res. Policy*, vol. 9, no. 2, pp. 56–71, 2018.
10. M. Agarwal, "Blockchain: India likely to see brain drain as 80% developers may move abroad," 2018. [Online]. Available: https://inc42.com/buzz/blockchain-india-likely-to-suffer-brain-drain-as-80-developers-prepare-to-move-abroad/

Nir Kshetri is a Professor of management with the Bryan School of Business and Economics, University of North Carolina at Greensboro. Contact him at nbkshetr@uncg.edu.

This article originally appeared in IT Professional, *vol. 21, no. 3, 2019.*

Silver Bullet Talks with Nick Weaver

Gary McGraw | Synopsys

Hear the full podcast and find show links, notes, and an online discussion at www.synopsys.com/silverbullet.

Nicholas Weaver is a staff researcher at the University of California, Berkeley's International Computer Science Institute (ICSI). He also teaches courses at Berkeley. Weaver joined ICSI in 2003 as a post-doc after earning a PhD in computer science from Berkeley. His research focuses on network security, worms, botnets, and other Internet-scale attacks. He also works on network measurement.

ICSI is a nonprofit computer science research center. How is it funded?
It's almost entirely grant funded. As a researcher at ICSI, I'm very project and grant focused, and this is why I am doing more lecturing at Berkeley, because as a lecturer, I don't need to worry about research grants.

What are your views on ICSI tech transfer into the world?
As a research lab, we like building things that work. For example, the Bro Network Security Monitor was developed at ICSI, and that's being commercialized right now. Ten years ago, there was the extensible open router project, and there was a significant attempt to tech transfer that.

There are also systems that we've ended up building that have monetization models that don't match industry, but are productized. The Netalyzr network analysis tool that we originally wrote in Java in the web browser now runs on Android phones. We keep that running because it pays us in research results. We are able to turn the service into publications, and therefore we have a monetization strategy. It couldn't actually work out in the real world, but works for us. And we end up supporting a large number of users that way.

That's good stuff. You and I seem to share the same skeptical stance when it comes to cryptocurrencies and blockchain. Can you briefly give us a synopsis of your recent Burn It with Fire webinar?
I've come to this after five-plus years of watching the field and occasionally publishing on it. What it comes down to is there's actually three totally separate concepts. There is the concept of the cryptocurrencies themselves. There is the concept of the public blockchains, and then there is the concept of the private or permissions blockchains. Now let's start with the latter.

What is a private or permissions blockchain? Simply an append-only data structure with a limited number of authorized writers: aka, a git archive. There is nothing fundamental in a private blockchain that hasn't been understood in the field for 20-plus years. It's just it has a buzzword that causes idiots to throw money at the problem. If you see a private or permissions blockchain project, it means either one of two things. Either it's a delusional piece of techno-utopianism, or somebody smart in IT knows that there are real problems with what data you store, or how you access it, data provenance, and all this other stuff, and has bandied around this buzzword because idiots up in management will now throw money at this person to solve the real, interesting, hard problem.

That's one of the three. What about the other two?
The public blockchains are a global data structure where the idea is there is no centralized point of trust, but anybody can append to it. Now these systems are, let's say, not actually distributed as advertised. The Bitcoin blockchain is actually effectively controlled by only three entities, but in an attempt to be distributed, there is this religious notion that distributed trust is somehow good in and of itself. The result is systems that are either grossly inefficient or insecure.

The biggest tool that's used for these systems is what is called "proof

About Nick Weaver

Nick Weaver is a staff researcher at Berkeley's International Computer Science Institute (ICSI). He also teaches courses at Berkeley. Weaver joined ICSI in 2003 as a post-doc after earning a PhD in computer science from the University of California, Berkeley. His research focuses on network security, worms, botnets, and other Internet-scale attacks. He also works on network measurement. Weaver holds a BA in astrophysics and computer science. His thesis work was on FPGA architectures, but he's focused on computer security since 2001. He lives in Berkeley.

of work." And proof of work is best described as "proof of waste." The idea is that for somebody to rewrite the history, they have to do as much useless work as was done to create the history in the first place. Now this is great if you do a lot of useless work, except then it's inefficient. If you make the system efficient so you do not do a lot of useless work, you run into the problem of not actually having any real protection.

For example, Bitcoin, since the proof of work is paid for by the newly minted coins, ends up using as much power as New York City. It's just an obscene waste of energy. At the same time, these distributed public append-only ledgers only have been useful for cryptocurrencies. Now it's time to address the elephant in the room; the notion of the cryptocurrency itself.

Right? Back to one. Here we go.
Cryptocurrencies don't actually work as currency. They are provably inferior and can never be superior to the alternatives for real-world payments, unless you need what is known as "censorship resistance." If I want to transfer you $500 by PayPal, or Venmo, or whatever, we have these trusted intermediaries called banks, and they make it relatively cheap. However, there is a problem. If I want to transfer $500 to you for drugs or the like, these central authorities don't like it.

The only way to do censorship-resistant transactions without a cryptocurrency is cash, and cash requires physical proximity and math. One million in US dollars weighs 10 kilograms. That's a considerable amount of stuff to be lugging around. What a cryptocurrency is, well, let's do a direct to peer-to-peer payment system so that there are no central intermediaries, but let's do it electronically. This has been used quite practically for drug dealers, extortionists, fake hitmen, and all sorts of things like that. But if I want to do any payment that one of the central authorities will process, the cryptocurrencies provably don't work.

Let's say I want to buy a couch from Overstock.com using bitcoins. I have to turn my dollars into bitcoins, because I don't want to keep it in bitcoins because the price is jumping up and down. That is expensive. Transfer the bitcoin. That is relatively cheap right now, but it's been upwards of $30 in the past. And then the recipient on the other side has to convert the bitcoins back into dollars. You have these two mandatory currency conversion steps for any real-world transaction, and even Overstock, the one public company that supposedly embraces cryptocurrency, only keeps a few hundreds of thousands of dollars' worth of cryptocurrency, with the rest converted to dollars.

Cryptocurrencies do not work for legitimate purchases if you don't believe in the cryptocurrency. But let us suppose you believe in the vision of the great Satoshi. Then you don't want to use cryptocurrencies either, because they're baked in with these monetary policies that are designed to be deflationary. The first rule of a deflationary currency is never spend your deflationary currency.

There is one aspect of cryptocurrency that I think people don't understand, and it is this notion of tethers. Can you talk about that for a second?
There is a way to make a cryptocurrency work. You have to have an entity that takes dollars and gives you crypto dollars at par, and vice versa, that will take the crypto dollars and return you dollars. This is called a "bank," and these are called "banknotes," and it's recreating the 18th-century banking system. This can work, but one of three things has to happen. One option is you have regulation and enforce money-laundering laws and everything else, in which case you have a system that ends up being no cheaper or no more expensive than Visa, or Venmo, or anything else. What is the point?

Option number two is you have what is known as a "wildcat bank." This is a bank that prints banknotes that are actually unbacked. And this is a term from 18th-century banking.

The third option is a Liberty Reserve where you actually do back up your reserves. You redeem your digital banknotes, but you don't follow the money-laundering laws, in which case you end up being a guest of the federal government for the next 15 to 20 years.

At the same time, the money that the average person had is tied up temporarily or forever when the Feds shut down the institution. Tether is a specific cryptocurrency that promises to be backed by dollars; they promise that there is this 1:1 ratio where you give them dollars, they give you tethers, and vice versa. The problem is this is almost certainly a wildcat bank because they managed to produce some 2 billion tethers in the space of a few months, and they are tied to a Bitcoin exchange that is otherwise cut off from banking. It may have been the direct reason why the Bitcoin price shot up so much.

Or they could be facilitating criminal money laundering, in which case those behind tether are liable to be guests of the federal government. This is, however, what actually enables most of the Bitcoin exchanges. Very few of the cryptocurrency exchanges actually are connected to the US banking system. You have Coinbase. You have Gemini, and you have Kraken (which should actually be shut down for other reasons of criminal activity, but that's neither here nor there). As for the rest of the exchanges, you can't actually transfer money into and out of them. These are where the hundreds and hundreds of different cryptocurrencies are actually traded on.

Tether has become this de facto reserve currency. If you look at Bitcoin trading volume, most of it is actually on tether-denominated exchanges and is not actually being exchanged for dollars, but these notional cryptodollars that may or may not be backed up, may or may not be a criminal enterprise—the flow just seems to continue on. It's really actually surprised me that it's lasted this long.

Yeah, it really is absolutely stunning this stuff. Thanks. That was extremely helpful. I think a lot of people need to have their eyes opened on this stuff, and you're one of the main people doing that.

I feel I have an obligation to. I kept looking at the field, and in the recent run up, I came to the conclusion that it's no longer harm-limited to a small population of self-selected believers. It is spilling out into the regular public.

Fortunately, I think the cryptocurrency space can die with proper application of regulation because of how the regulations already are, but it's become important for me to advocate for the need to clean up the space in that cryptocurrencies don't provide benefit to society. They don't provide benefit to all of us who aren't interested in committing crimes, but they do enable these problems. I think it is important to speak out. Another thing is the amount of scams in the space is just incredible.

Effectively every initial coin offering these days should be called a scam, because it is an unregistered security and wouldn't even pass the laugh test on Shark Tank. And we have got these people hyping smart contracts. Most of the cryptocurrency community seems intent on speed-running 500 years of economic history for choosing their bad ideas, but smart contracts are actually a new bad idea. The idea behind a smart contract is that I write a program that is not really a smart contract, it's a finance bot, because if it's a contract, you have this exception-handling mechanism called a judge in the legal system.

If I can walk up to a smart contract, say "Give me all your money," and it does, is that even theft? Well, it would be theft in the real world because we believe in justifying things, and this exception-handling mechanism of the judge and jury and all that. Smart contracts are instead—let's take the idea of a contract that is standardized and written in a formal way, it's called "legalese," and instead, rewrite it in a language that is uglier than JavaScript and has all sorts of pitfalls for programmers, eliminate the exception-handling mechanism, and then require that the code be bug free.

Except it's not bug free.

Oh, it's so amusingly not bug free. I like to use three examples. The first is the DAO, the Decentralized Autonomous Organization. The idea is, let's create a self-voting mutual fund for how we can invest our cryptocurrency in other projects. Now that there's actually nothing to invest was neither here nor there, but around 10 percent of all Ethereum at the time ended up in this basically self-creating, self-perpetuating, not-quite-a-Ponzi Ponzi scheme.

This was all fine and good until somebody noticed there was a reentrancy bug that allowed them to say, "Hey DAO, I am an investor. Give me all my money." And in the process repeat the thing as, "Hey DAO, give me all my money." And because there was a transfer then update, and you could re-entrantly call this code, it basically sucked all the money out.

The problem is, well, the money that was stolen mostly belonged to the people who came up with Ethereum in the first place. They basically did a code release that changed it and undid history. Their notion that code is law and there is no central authorities and no way to undo things was revealed to be a transparent lie when it's their money on the line.

Exactly.

So that's number one. Number two is the Proof of Weak Hands explicit Ponzi Scheme. Version 1.0 collected several million bucks before one bug locked it up so nobody could transfer any more money into it, and another bug allowed somebody to steal all the money in it. I think they're up to 3.0 now, which has yet to have a fatal bug, but we'll see how long that lasts.

Finally there is the Parity multisig wallet. One of the problems of cryptocurrencies is you can't actually store your cryptocurrency on an Internet-connected computer because if somebody gets onto your computer,

This article originally appeared in
IEEE Security & Privacy, *vol. 16, no. 4, 2018.*

they get your private key and steal all your money. We actually had this happen to us in the early days of Bitcoin, and if security researchers can't use Bitcoin on an Internet-connected computer, nobody can. The idea is, let's make it a two-party check system. We will have three private keys, and you have to use two of them to transfer the currency.

This gives you good controls if you can theoretically maintain at least two of your cryptographic keys. Some systems, like Bitcoin, offer it as a primitive. For Ethereum, it was built as a smart contract on top of things. This was the Parity multisig wallet, which collected some hundreds of millions of dollars, including an ICO by the guy behind the Parity multisig wallet. Until somebody noticed that there was a bug where you could go up to one of these wallets say, "Hey, wallet. You belong to me. Hey, wallet. Give me all your money," and started cleaning these out. And the only reason this wasn't a $150 million theft is somebody else noticed that this was going on, stole all the money first, and then gave it back to the victim once the victim had upgraded code.

Unbelievable.

Which gets better. Now there's the upgraded wallet code. For efficiency, everybody refers to the same wallet contract, and there was a bug in this contract. Some random loser came along and said, "Hey contract. You belong to me now," and the contract said, "Okey-doke. Yeah, I do." Okay, oh crap. This shouldn't have happened. "Hey, contract. Kill yourself." The contract committed suicide, and now $150 million worth of cryptocurrency is locked up and effectively inaccessible unless the central authorities, that aren't supposed to exist, change the code to unlock this. We're not done yet. The pièce de résistance.

The lead programmer and shining light behind this fiasco is the guy who invented the programming language in the first place. The problem is these things are designed to be non-upgradeable, but there are hacks that allow you to update them. If your money is tied up in somebody else's contract because their contract is the service, you have a choice. Either that contract has to have been bug free when created, not good, or that contract has to be upgradeable, in which case you have to trust that they upgrade the contract properly and don't cause damage or work against you in the process.

You have a central authority again.

You have a central authority. For example, there was a bug discovered in some of these smart contracts that run these ICOs, where somebody was able to create, what was it, 200 billion new tokens? Well, the people in charge of that particular smart contract were able to undo the process, but that means also if they can destroy the hack-created tokens, if you're invested in them, they can destroy your tokens too if they feel like it.

You have to trust them.

This is the ultimate irony in all these systems—their belief in this mantra that lack of trust and decentralization are good in and of themselves, ignoring the huge advantages you get with just even the slightest of smattering of centralized trust. Yet they end up building systems that aren't even decentralized. They build things that are orders of magnitude less efficient than they could be, but which have central authorities and aren't distributed anyway.

I think the real design decision was, "I would like to have all the trust belong to me."

No, the cryptocurrency community truly believes in this idea of decentralization; that you should have to trust nobody.

They're just bad at implementing it.

They don't understand the costs involved in that, and they cannot seem to ever implement it that way anyway.

All right, so onto a very personal issue. You suffer from depression that's treated by therapy and medication, and you talk about that so others can benefit from the good aspects of treatment and therapy. Tell us a little bit about that.

I've basically had in my life multiple depression meltdowns, and therapy and drugs saved my life twice as a student. And both times, after about a year, I'd just go off the medication, and a couple of years later the same thing would happen again. Just after the third incident, I realized that I didn't want to repeat that mistake.

So, when I'm teaching students, every semester I include in my first slide deck, the notion that yes, I've been there. I've done that. This is not good. There is help available. Every semester at least one student has proven that it's been worthwhile and they'll come up to me afterward.

Super important work. Last question, what is your favorite fiction book or your favorite fiction book you're reading at the moment?

Let's just say I'm a huge fan of *The Laundry Files*.

The Silver Bullet Podcast with Gary McGraw is cosponsored by Synopsys and this magazine and is syndicated by SearchSecurity. ■

Gary McGraw is vice president of security technology at Synopsys. He's the author of *Software Security: Building Security In* (Addison-Wesley 2006) and eight other books. McGraw received a BA in philosophy from the University of Virginia and a dual PhD in computer science and cognitive science from Indiana University. Contact him via garymcgraw.com.

Co-creation and Risk-Taking—In Pursuit of New Technology for Human Augmentation

An Interview with Pranav Mistry

Marc Langheinrich
Università della Svizzera Italiana (USI)

Nigel Davies
Lancaster University

Marc Langheinrich and Nigel Davies interview Pranav Mistry, Global Senior Vice President of Research at Samsung, about his views on the field of human augmentation.

Creating new technology for augmenting humans represents a major challenge to both industry and academia. To explore how the field has developed over recent years, as well as discuss future trends, we sat down with Pranav Mistry, Global Senior Vice President of Research at Samsung and director of its "Think Tank Team," an interdisciplinary group that aims to create Samsung's "products of tomorrow." Examples of Pranav's work include visionary contributions such as the SixthSense device—a wearable gesture interface that is the subject of one of the most watched TED talks of all time—as well as commercial successes such as Samsung's smartwatch, the Samsung Gear. The idea of augmenting humans has been at the center of Pranav's work for many years, so we were excited when we got the chance to interview him for this special issue.

Looking back to 2008 when you started to work on SixthSense, how do you think the field of "augmenting humans" has changed?

A couple of things have changed since the early days of research into human augmentation.

First, at that time, academia and industry had very different ideas. In academia, many of us were thinking about the idea of augmenting humans, while in industry this wasn't much of a topic at all. When I look at the field today, industry and academia are moving in the same direction, looking at the same problems—both are thinking much more about bringing human augmentation to the mass market.

The second thing that has happened since 2008 is that, because of the new form factors of computing, everyday users have started accepting the possibility of human augmentation. Of course, for many years, maybe even several hundred years, we've been "augmenting" ourselves with simple things, like reading glasses or a watch. I remember that during my school years I used to wear a Casio watch that allowed me to store my phone numbers on my wrist. Nowadays, not only geeks like me and you do this. "Regular" people also think that in the future "I might live like that" or "I might wear that kind of device," or even "I might directly connect to computers using my brain."

This new trend of accepting human augmentation isn't actually coming from either academia or industry; it's coming from the media, from science fiction movies and stories. And that's helpful for all of us, because unless some people accept the fact that it's okay to add something to their bodies, to their cognition, there won't be support for computers augmenting our memories, or our intelligence, or our reasoning.

This new trend of accepting human augmentation isn't actually coming from either academia or industry; it's coming from the media, from science fiction movies and stories.

Back in 2008, several companies and small startups were doing something similar to what we're doing right now—creating this world of augmented reality (AR) and virtual reality (VR), and new kinds of augmentation technologies. But the investment industry wasn't taking them seriously at all. Now, the situation is flipped upside-down—if someone comes up with a completely different augmentation technology, even if it has nothing to do with currently accepted technology, investors will support it, thinking there might be a market for it in the future. From a technologist's perspective, acceptance of novel augmentation technology as a potential big thing is a huge and positive change.

Google Glass attracted quite a lot of negative press in the past. Do you think this was because of something inherent in the technology or would the reaction be different today?

I would say that a fair debate is needed, at the individual as well as the societal level, on what's right or wrong with any new technology. This debate is probably going to emerge as a result of trial and error. I don't think a social scientist or psychologist can simply declare that people should or not should not accept the technology. Take, for example, self-driving cars: it seems that, perhaps due to their depiction in sci-fi movies, people already accept this technology. But is the technology safe? Is it for everyone? You can't have an informed debate until there are enough self-driving cars on the road.

I wouldn't say Google Glass was a failed technology; it was a necessary intermediate step that allowed the public to understand how such technology works and could fit into their lives. People often react negatively to a new technology but slowly come to accept it as they see its benefits. When we started research on Samsung VR and told users to insert their phone into the headset and put it in front of their eyes, their initial reaction was "Are you crazy? What are you talking about?" In time, people recognized its value for entertainment and other kinds of applications. And, of course, how a technology will play out in the market isn't exactly what industry expects. VR is becoming less of a virtual reality experience and more of an "other reality" experience in which people want to be transported to another place, such as the other side of the world. Technology changes the perspective of both the user and the maker, and a period is needed to assess those changes.

Has technology in the human augmentation space evolved as you foresaw when you were doing your PhD?

Some aspects of technology have seen much faster progress than I expected, while others have been much slower. In particular, I thought software development would be well behind hardware development, but it has been the other way around because today there are a lot more experts on machine learning, AI, and computer vision algorithms. With respect to hardware miniaturization, there isn't much difference between what we used nine years ago and what we're using today. The Galaxy phone I have now is similar in size to the one I had in 2008. Hardware hasn't changed as drastically as I anticipated.

I think there are a couple of reasons for this. One is that, when thinking about future hardware, we quickly forget about the fundamental limits of physics, which play a crucial role in the hardware field. Another is that the emergence of the cloud has enabled software development to accelerate. Although we didn't know it a decade ago, the infrastructure needed for future applications was being prepared.

Right now, if you wanted to make a memory or visual augmentation system, hardware is the limiting factor. We can explore different software solutions—a better UX or maybe a contextual interface—but physical silicon is harder to change. You can't do trial and error with hardware as much as you can with software.

Would it be fair to say that it's been a long time since we've seen hardware in the human augmentation space that's really been transformative?

Yes, I think so. It takes time to develop hardware—there are many steps—but the good news is that pipelines are in place from the labs to end users. While hardware is still the limiting factor, we can bring innovations to the market today much faster than we would have been able to, say, five years ago. Industry has accepted the first step in this pipeline—that this technology has mass market potential—so we can now develop a market strategy much more quickly. Also, it's not just about the computing or electronics industries anymore; the automotive industry is talking about more cameras and sensors, the fashion industry is exploring new kinds of smart accessories and smart garments, and so on. And that's what's so exciting: the everyday world is starting to get augmented.

Right now, if you wanted to make a memory or visual augmentation system, hardware is the limiting factor.

So while the hardware side will continue to innovate, the rate of progress will be somewhat bounded compared to the software side?

Yes, definitely. Even now, software improvements are defining what new hardware improvements are needed because the software side is much further advanced. For example, Apple, Google, Microsoft, and the like are building chips just for machine learning or computer vision; that wasn't the case several years ago. The big industry players have already started thinking about the coming new world of human and digital augmentation and they want to be prepared. In the early years of computing, hardware defined the software, which in turn defined the user experience. Today, popular culture like sci-fi movies and games is defining new user experiences, which we're implementing in software; we then create custom hardware to enable this software-defined vision.

Earlier you mentioned application areas for augmenting humans, such as memory or vision augmentation. Do you see a "killer app" in this space?

My thinking is that, as been the case with many new things, consumers will readily accept augmentation applications in a noncritical area like gaming. Industry, on the other hand, is always a bit more skeptical about new technology, and hence about which particular area to explore. For example, Microsoft's HoloLens or Google Glass are finding a home in niche B2B industry applications because it's a much more controlled environment. Even at the device level, we can see this exploration at work. For example, my Galaxy phone has Bixby Voice and Bixby Vision, but my primary method of communicating with my phone is still the touchscreen because it understands what I want 100 percent of the time. Soon, however, I might not need to tap on the screen anymore, and that's awesome. I believe this will happen with augmentation technology as well: new technology will coexist with traditional technology until users feel comfortable with it, and then they'll make the switch.

So I believe the proving grounds for human augmentation technology will be either noncritical areas such as entertainment or very controlled environments in critical areas such as business, where the technology does limited things but does them perfectly. Enterprise applications are especially promising for exploring augmentation technology because this world isn't fully open. At the consumer level, the technology initially will mostly be for fun. Take AR: most people use it to augment photos on sites like Facebook to, say, put a cat on their head or to add virtual sunglasses or a smiley face.

So if we start out with "fun" things first, what do you think this will morph into?

When we started working on Gear VR, we thought it was all going to be about *virtual* experiences such as immersive gaming or exploring imaginary worlds. We didn't know which particular aspect of the technology was going to be most important to users. We found out that what people wanted more than anything else was to use the device to "teleport" to another part of the *real* world. For us that revelation opened up an entire ecosystem of new devices and 360-degree cameras to explore, and new research goals like the network capabilities to support streaming high-definition live video. Now when we introduce a technology we tell users that it can do this, this, and this, but we leave it up to them to decide what they would like to do the most and then follow their lead.

I believe the proving grounds for human augmentation technology will be either noncritical areas such as entertainment, or very controlled environments in critical areas such as business.

Let me tell you an interesting story. A couple of years ago, I was in a restaurant and at a table next to me was a group of young teenagers all using their smartphones' heart rate monitor. Such an app is targeted at older users, so out of curiosity I asked what they were doing, and they said they were playing a game in which the person with the highest heart rate had to pay the bill. These teens had thus taken something created for a serious use case and repurposed it for something fun instead, and in so doing were extending the application spectrum of this kind of technology. As this story shows, it's impossible to guess all the uses people will make of the things we create.

At Samsung, we always say that we listen to our customers—we listen to what they want, we create a new technology, we listen to the response and change the technology, and so on until we get it right. This is essentially a customized process for designing the future. When Ford launched the Model T, there was one model and it was only available in black—everyone had the same car. That doesn't work anymore: today people want devices that adapt to *their* needs. And that's how it should be, since our needs are different. My father is an architect; his memory or visual augmentation needs are completely different from mine. Even if everyone eventually uses augmentation technologies like they use smartphones today, it's unlikely they'll be "augmented" the same way.

I thus see a future with a proliferation of customized augmentation technologies. One person might have an app to help remember names or phone numbers, while another might use one to help translate Chinese to Japanese during trips to Japan. At the same time, creating a technology that can satisfy the diverse needs of millions of customers at an efficient production scale of billions of units is a huge challenge.

I should mention that a friend of mine in academia is interested in personalized artificial intelligence. His idea of AI isn't a single "Watson" that serves everyone equally. Instead, your personal assistant grows with you, has the same social and cultural experiences, and shapes itself in line with your personality. I believe the same "one size doesn't fit all" notion equally applies to human augmentation. It also has interesting implications: virtual agents with their own personality might have problems communicating with each other or different users, leading to all sorts of misunderstandings!

Could you talk a little bit about your own current research, and that of Samsung as well, in trying to realize this vision?

Samsung, at its core, used to be a hardware-heavy company: we understand silicon much better than anyone else in the world. And as I mentioned before, it takes time to grow silicon—it's like a tree; it doesn't grow overnight. It takes a lot more time to develop a chip than to convert an algorithm into running software. If software doesn't behave properly, you can change some lines of the code, but if hardware has an issue, you have to rebuild it. And then there are the physical limits of the atoms compared to the bits. Samsung's research focus is "being ready for the future." It sounds simple, but it's hard because no one knows where we're heading and so we have to make some big bets in the R&D area. Some of my work focuses on thinking about the future and what its silicon requirements might be so that we don't lag behind the software industry and what consumers want.

Earlier in this interview I talked about how, as a first step, we have to "test the waters" in the real world—you can't just skip straight to wide-scale deployment of augmentation technology. That step one is also my focus: to see if and how society accepts a new technology like Google Glass and what kinds of problems they have with it in terms of privacy, security, or just how it looks or feels. So, personally, my research extends across the entire development spectrum because we have to take risks in this industry to move forward. We're like a startup: notice that Samsung is among the first to try out new technologies on the market, from curved screens to VR to 360-degree cameras.

Throughout our discussion has been this thread that software is moving very quickly—much more quickly than any of us predicted. Hardware moves more slowly, and because of that, and because you want the silicon to be in place to support new software, you must maintain a long-term perspective. As part of that process, you've got to put stuff out there and see what sticks. Is that a fair summary?

In one way, yes. However, the last point is really is about co-creating the future with our users; it's not just a matter of putting something on the market and seeing how it goes. No matter how good an initial product is, it isn't perfect; it's not "tomorrow" yet, but we're almost there—a minute to midnight. That's what drives Samsung: let's find out what works and what doesn't, and then let's make it better.

So it's important to do co-creation and co-design early?

Absolutely.

Finally, when it comes to taking risks, what would you like to see from academia?

At the start of this interview I said that academia and industry are moving in the same direction, and this is both good and bad. On one hand, industry and academia can benefit a lot by collaborating and sharing a common perspective. On the other hand, we might overlook something interesting that might have been.

There was a time when new ideas about technology mostly originated in academia, but now the situation is almost reversed.

Academics have more freedom to explore; they should go wild and try out new things! We're seeing fewer contributions from academics and more from industry. There was a time when new ideas about technology mostly originated in academia, but now the situation is almost reversed, with industry saying, "We have this technology—can you academics do something with it?" I feel academia should again become a little bit more dominant in telling industry where they see the future going. The key challenges academics can focus on, which is harder in industry, are those we discussed before: testing people's experience, what works and what doesn't; seeing, even in a small way, what users like and what they don't; understanding what people expect the new world to be like.

Ultimately, industry and academia must work together to solve the bigger societal problems, not just technical issues related to software and hardware. Because the main question about augmenting humans is not about what we *can* do but what we *should* do, which will ultimately determine whether the technology succeeds or fails. It's this moral aspect in which academia needs to play a major role.

So it sounds like a call to arms for the world's philosophers and ethnographers?

I think so, because they're critical to realizing this vision.

Thanks a lot, Pranav, for taking time out of your busy schedule to speak with us.

ABOUT THE AUTHORS

Marc Lanheinrich is a professor in the Faculty of Informatics at Università della Svizzera Italiana (USI) in Lugano, Switzerland, where he heads the Research Group for Ubiquitous Computing, and Editor in Chief of *IEEE Pervasive Computing*. Contact him at langheinrich@ieee.org.

Nigel Davies is a professor of computer science at Lancaster University and Editor in Chief Emeritus of *IEEE Pervasive Computing*. Contact him at nigel@comp.lancs.ac.uk.

This article originally appeared in
IEEE Pervasive Computing, *vol. 17, no. 2, 2018.*

Compressing VR: Fitting Large Virtual Environments within Limited Physical Space

Khrystyna Vasylevska and Hannes Kaufmann
TU Wien

Ideally, a virtual reality (VR) system should connect a real person to a computer simulated world, allowing the system to fully substitute the real world and its rules. Like the Holodeck featured on the TV series *Star Trek*, such a system should be able to provide an interactive, tangible virtual world that the user can explore without restrictions within a real room. One of the first ways someone might attempt to explore such a world would be to walk around. Nonetheless, as a result of restricted physical workspaces and technological limitations, the free and unlimited exploration of an arbitrary large-scale virtual environment (VE) is not possible in practice. We could rely on walk-like gestures or use additional devices to allow users to travel through VEs, while their physical locations do not change. However, real walking in VR provides important vestibular and proprioceptive cues that positively impact higher mental processes and improve the illusion of reality.[1]

In this article, we provide an overview of the existing approaches and techniques for enlarging the walkable virtual space. We specifically focus on the methods that use spatial manipulation for spatial compression, as it is one of the most promising, but underexplored methods for nonintrusive user redirection in a limited physical space. Researchers have developed several techniques to address the problem of free natural locomotion in VEs within an available real-world workspace. We distinguish the following types of spatial compression methods:

- basic reorientation,
- sense manipulation,
- rendering manipulation, and
- 3D scene manipulation.

All of them target the highest possible compression factors for any virtual space, and each has its own benefits and challenges.

Basic Reorientation

The most basic approach is to stop users at the boundary of the tracked space and ask them to return to its center and continue from the same point in the VE.[2] Rotation can also be instantaneously introduced based on the user's position in the real world.[3]

These basic approaches interrupt the VR experience and thus might adversely impact important characteristics of it, such as immersion and a sense of presence in the VE. More intricate methods of redirection exercise unperceivable manipulation, while the rendering and the user's immersive experience remain intact.

Sense Manipulation

One class of techniques known as *redirected walking* employs sense or orientation manipulation.[2] These methods build upon the principle that, during the multisensory integration process, visual cues are usually weighted as more accurate and therefore more important for orientation than other senses such as proprioception. Redirected walking uses the concept of camera manipulations based on gains. The user's dynamic motions are scaled according to the defined gains and then mapped to the translation and rotation of a virtual camera within a VE. The user reacts to the changes in the virtual camera's pose and adapts his/her motions accordingly, which in turn lets us keep the user within the real workspace.[2]

It is also possible to continuously apply the additional rotation. A generalized version of this

2469-7087/19 © 2019 IEEE Published by the IEEE Computer Society September 2019

approach is called the *circular algorithm*,[3] which mainly consists of two main types of manipulation and their combinations. The first keeps users on a small circular trajectory, allowing them to diverge in any direction. The other constantly redirects the user to the center of a big circle when the user performs a rotation. The goal is to make the additional rotation imperceptible to the user. For example, it may be applied when the user is performing fast head motions trying to follow a fast-moving object. This approach is referred to as the *distractor technique*.[4]

Human sensitivity limits the extent to which we can apply manipulations in virtual spaces[5] because such manipulations of primary senses should remain unnoticeable to users to minimize the possible adverse effects. Hence, sense manipulation still demands a considerably large real workspace. For instance, for users to continuously walk along a straight path in a VE with a curvature gain requires a squared workspace of almost 500 m^2.[5] Research has shown however that the radius might be decreased by a factor of two if the curvature gain is accompanied by translation gain.[6]

In practice, redirection by sense manipulation works well for moderately paced users who try to follow the planned path, but it can fail in other circumstances and scenarios. Therefore, sense manipulation is most suitable for outdoor open VEs where the virtual path might be easily adjusted to fit the real workspace. Nevertheless, the use of sense manipulation requires fine-tuning and extensive testing of each particular VE, and such testing should account for some unexpected user behavior.

Rendering Manipulation

Qi Sun and his colleagues proposed a novel rendering approach to spatial compression.[7] Their technique consists of a planar mapping of the constrained walking path with a custom reprojective rendering that is capable of wrapping an arbitrary VE into any real-world workspace. The obvious benefit of this approach is its flexibility. However, their method distorts the VE's visuals and makes it difficult for users to estimate the scale and exact shape of the environment.

Because this technique alters the user's perception of the environment, it needs to be explored further. Nevertheless, this approach could also be successfully applied to outdoor virtual scenes that involve content that is less sensitive to distortions.

Scene Manipulation

Unlike the previous approaches, virtual scene manipulation has an enormous potential to increase the compression factor of VEs without the need to manipulate the users' senses in an unnatural way. The core approach in scene manipulation is to have different parts of a VE share the same real workspace. To do so, some parts or elements of a VE are relocated, overlapping in the real-world space based on the users' actions. Most importantly, these changes occur without the users noticing.

One basic spatial manipulation approach involves the use of *deterrents*. That is, during runtime, objects are inserted into the VE that users must avoid walking through, such as roadblocks, which forces them to take an alternate route within the environment.[8]

Other approaches go further, changing the VE's configuration more drastically while users explore the virtual space and perform tasks.

Change Blindness

Change blindness is an entirely different approach to spatial compression wherein the system or specific task distracts users so they fail to notice large changes in VEs.[9] In the first study,[10] users were asked to perform a task that required they turn their backs to a door. While the users were distracted, the door's location was moved to a different wall in the virtual room (see Figure 1a). An interesting outcome of the study was that, after exploring the virtual building, the study participants were able to draw a map of the environment despite substantial spatial manipulations.

A second study tested more significant scene modifications based on change blindnesss.[11] In this second study, the entire wall containing the door was moved several meters away from its original position; this change significantly enlarged the room in order to return users back to the real starting point. Such an approach is most suitable for environments that contain regular structures, although generalizing and expanding the approach to arbitrary spatial arrangements is still problematic.

Impossible Spaces

Another method to compress VEs is the use of *impossible spaces*.[12] This approach increases the amount of walkable space by making separate rooms overlap and partially share the real space with one another. There are two possible implementations of impossible spaces. One involves expanding the space available in adjacent rooms by moving their shared wall and increasing the overlap (see Figure 1b). At the same time, the outer walls, doors, and the connecting corridor do not change. The other implementation involves increasing the overlap

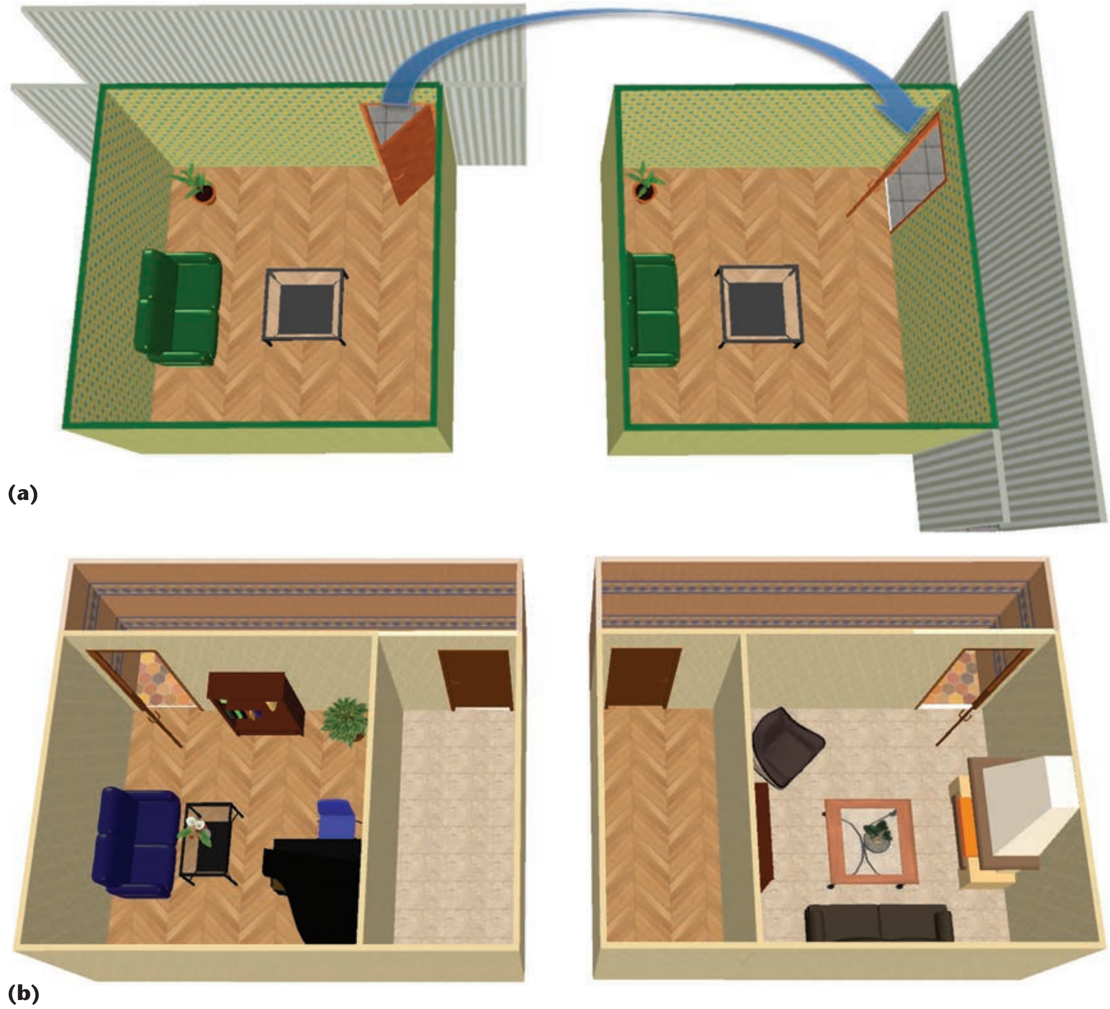


Figure 1. Spatial manipulations: (a) In the change blindness approach, the door is relocated in the virtual environment (VE) when the user is distracted by a task.[10] (b) The impossible spaces approach lets us extend a room setup with 50 percent overlap. The wall between the rooms is relocated based on the users' actions in order to enlarge the room they are about to visit using the overlap area.[12]

by bringing the two rooms closer to each other to minimize the space needed for them as well as the length of their connecting corridor. A study on impossible spaces showed that when blind walking between the identically placed objects in both rooms nonnaïve users failed to estimate the actual distances between the rooms correctly. That result suggests the use of impossible spaces efficiently increases the sizes of walkable virtual environments.

We preformed a follow-up study for impossible spaces showing that, by changing the complexity of the corridor, it is possible to increase the amount of unperceived overlap.[13] In this case, we define the complexity by the corridor's length and the number of corners in it. We used an expanding implementation of impossible spaces and explored whether the overlap perception depends on the corridor that connects the rooms. As in the earlier study, we used blind walking as a measure. Figure 2 illustrates the three types of corridors we designed: a simple corridor; a U-shaped corridor, with which we extended a simple corridor an additional 10 meters, detaching it from the rooms' perimeter; and a C-shaped corridor, which we extended with another 10 meters and four additional turns.

Although the simple length extension did somewhat decrease the users' overlap perception, our results showed that it was not particularly efficient in terms of the use of available space. However, the more complex C-shaped corridor substantially impacted the users' spatial perception when compared with the simple and U-shaped corridors. The estimated distances between the rooms in this case suggested that the rooms were far apart from each other. Moreover, some of the participants also stated that the rooms were not aligned.

In later work, we further delved into the corridor-dependent effects on spatial perception by addressing the corridor configuration parameters

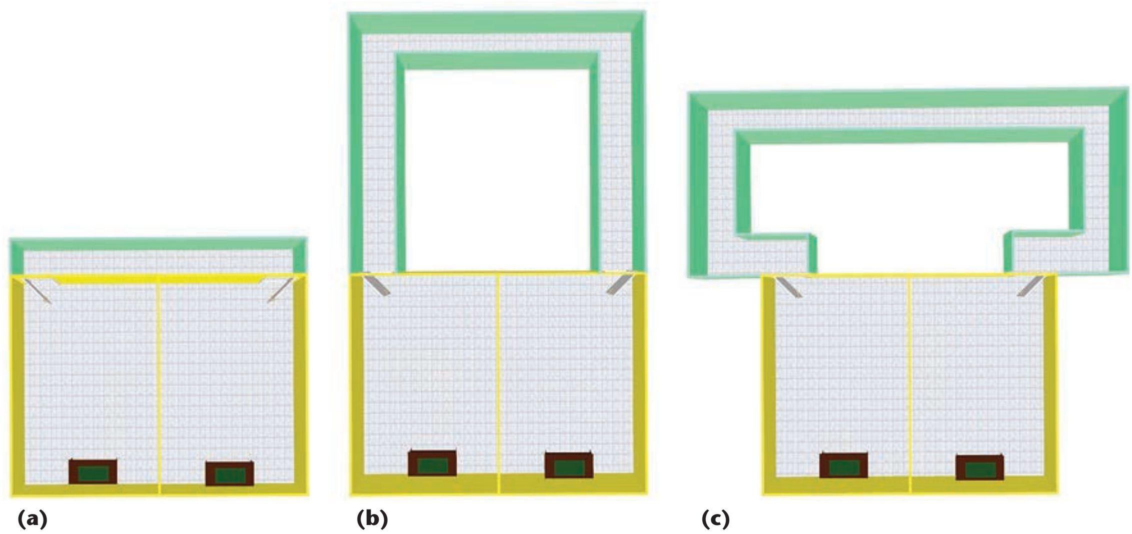


Figure 2. Virtual layouts with different corridors: (a) a simple short corridor, (b) a U-shaped corridor, and (c) a C-shaped corridor. The overlap was implemented by moving the wall between the rooms.

and geometry.[14] Furthermore, we diverged from the simple right-angled geometry. Instead, we used smooth curves and scrutinized their effect on spatial perception. We used two rectangular rooms of identical sizes that were aligned and overlapped by 50 percent throughout the experiment and focused only on corridor configuration. We hypothesized that the spatial perception in self-overlapping VEs might be influenced by the following properties of the connecting corridor:

- the number of corners,
- the sequence of corners,
- the positions of the corridor endpoints (doors) relative to the overlap zone, and
- the path's symmetry or asymmetry.

Based on these criteria, we created nine right-angled layouts, five of which were symmetrical and four asymmetrical. Figure 3a shows the right-angled asymmetrical layout. We also created a second set of layouts where the right-angled corridors were substituted with curved versions and tested this set separately. In this second set, we eliminated the corners and straight parts of the corridors that could be used as landmarks or for directional hints. Our objective was to see whether users would still perceive the room alignment and overlap in the same way and to evaluate the potential use of curved paths for spatial manipulations.

In addition, we assumed that asymmetrical layouts might feel different when participants walked in alternating directions. Therefore, we had the participants explore such layouts twice, in clockwise and counterclockwise directions. To measure the participants' spatial perception, we introduced a new approach: interactive visual reconstruction using semitransparent representations of the rooms (see Figure 3c). We also explained to our participants the possibility of the overlapping, adjacent, and completely detached rooms, challenging them to estimate the original room arrangement in each case separately.

The study results confirmed the importance of all the corridor parameters we have discussed here,

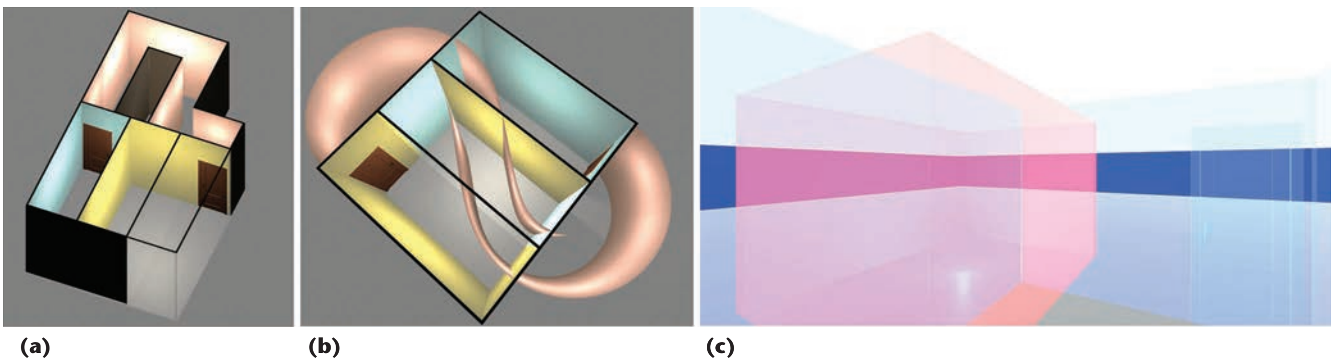
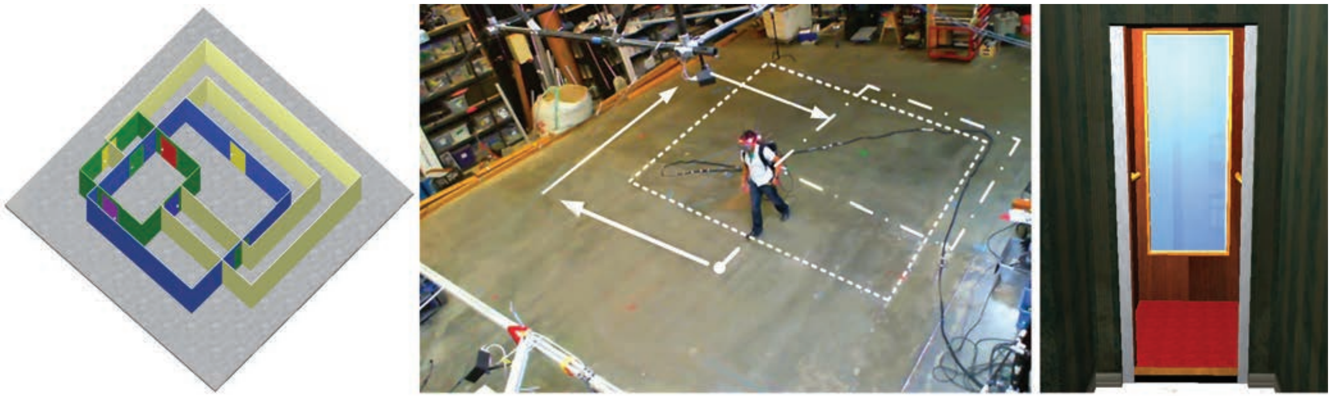


Figure 3. Experimental environment on the use of corridors in impossible spaces: (a) 3D models of symmetric and asymmetric right-angled layouts and (b) 3D models with curved corridors. (c) During task performance, participants were shown semitransparent representations of the rooms.



(a) (b) (c)

Figure 4. Flexible spaces: (a) a basic procedurally generated VE, (b) a user exploring the flexible spaces in the VE, and (c) an elevator extension.

the presence of distortions in the spatial perception, and differences in the perception of an asymmetric layout depending on the walking direction. Our results also suggest that participants were still able to perceive the overlap area and room alignment when they walked right-angled corridors.

The layout set with curved corridors provided an increased variation in estimated spatial arrangements and caused the participants to estimate larger distances between rooms compared with the right-angled set. The results indicated that in many configurations the participants believed there was space between the rooms. Unlike the right-angled layouts, some participants also asked whether the rooms had been rotated, which suggests a perceived change of room orientation.

Overall, the best results in both studies were achieved with the S-shaped corridor (see Figure 3b), which reliably created a long distance between the rooms. The S-shaped corridor was also the most space efficient because of the triple overlap as it passed directly through the area where the rooms overlap.

Earlier studies have confirmed distortions in spatial perception for larger real scenes, but to the best of our knowledge, our study is the first to directly observe a similar effect for small-scale self-overlapping VEs. Based on the obtained results, we suggest considering the parameters of the path that connects different spaces when designing impossible VEs. If possible, loop-like paths should be avoided as they might increase the perceived overlap. Meanwhile, the corridors that change the turning directions seem to be more realistic and decrease the overlap. The positions of doors relative to the overlap also matters, and it is best to position them as far from the overlap and each other as possible. The use of asymmetric corridors also proved to be efficient. However, the walking direction and placement of the elements that change the corridor's direction should be taken into account.

Flexible Spaces

The *flexible spaces approach* is one of the first attempts to merge several techniques. Our approach is based on the assumption that detailed spatial knowledge might be useful for navigation but is not necessary for all environments, particularly those that focus on information and content or impression and experience. A perfect example of such real-world settings is a large museum with signs that substitute the map of the building or the insides of a pyramid where loss of orientation is part of the experience.

The flexible spaces algorithm also relies on the fact that cognitive maps are often distorted, sometimes to the degree that they cannot be represented by images.[15] These distortions originate in the hierarchical structure of the cognitive maps and mental heuristics that help us to remember information about the environment. Thus, human perception gives us a way to create a new class of information- and content-oriented environments that provide consistent connections between their parts (predefined bidirectional links between the rooms) but that modify the details in between with a changeable architecture.

Our algorithm creates a procedurally generated self-overlapping and self-reorganizing dynamic VE that automatically regenerates the environment within the available workspace. In this approach, we united change blindness and impossible spaces, taking them to the extreme by allowing constant restructuring of the VE. Unlike previous work, our version of change blindness is task independent. The flexible spaces approach maintains the connections between the parts of the VE but does not repeat the layouts. The changes in the layout occur as soon as the user leaves a room or a corridor, and they are occluded by the other elements of the VE. Figures 4a and 4b show a procedurally generated layout for a VE with two rooms and a user exploring it. (See earlier work for a detailed explanation of the flexible spaces algorithm.[16])

The constantly changing nature of the algorithm prevents users from building up spatial knowledge and forces them to rely on other means for orientation. Following the museum metaphor, we introduced room-to-door color coding. For example, a red door always leads to a red room, making it content independent.

In our pilot study, we demonstrated that spatial overlap could be efficiently used in cases where it is not necessary for users to learn the spatial arrangement. Our test participants perceived the VE as something possible in the real world, which demonstrates the benefits of spatial manipulations for efficient workspace usage.

Another advantage of the flexible spaces algorithm is its versatility. It can be used in the originally proposed version or to generate unique, single-use layouts for each session. The algorithm supports an unlimited number and different sizes and shapes of rooms or other confined spaces, and it can easily be adapted to different room designs. Unlike other techniques, the flexible spaces algorithm guarantees unlimited walking with successful redirection and undetectable spatial overlap of up to 100 percent. In a case with a particularly dense spatial arrangement, it is possible to extend the environment to different levels with portals, flying, or a haptic elevator simulation (see Figure 4c).[17]

Challenges

Spatial manipulation still requires a rather large real space to create a believable VE. At the same time, our experience with flexible spaces and self-overlapping architectures suggests that users might consciously accept spatial manipulations. However, some users might also find the concept of an unrealistic architecture to be disturbing. Moreover, there might be an unexplored spectrum of new rules and techniques that users might consciously accept. As a next step, we plan to evolve the flexible spaces algorithm to accommodate curved geometry. That, in turn, might improve the compatibility with rotation and curvature gains. As for the existing methods, we consider combining multiple existing nonintrusive approaches for real walking support into a single ultimate technique to be one of the hardest tasks in achieving more efficient virtual space compression. Although some attempts have already been made, no perfect technique has been found yet. There are still open problems with large open spaces and support for a completely free exploration within a limited real workspace. To complicate matters further, the various types of VEs with real walking support are not universal and often require adaptation to specific real-world workspaces.

Another challenge for VR systems with large workspaces is estimating how many people a workspace could fit. Moreover, how do we support the simultaneous free exploration of multiple users within the same VR system? For that, we need fast, reliable, and smart path-prediction algorithms that take the user's behavioral specifics into consideration and novel methods to effectively counter any unexpected user behavior.

At this stage, VR researchers and developers should continue to explore and learn to exploit the limits of human vision, perception, and cognition in close contact with psychologists. Unfortunately, a large gap still exists between experimental psychology that uses very simple setups and stimulus and the demands of the striving field of VR. This gap needs to be bridged in order to keep pace with VR technology.

Lastly, the spread of consumer hardware is finally opening up possibilities for studying human adaptation to VR over a large population of users, but it raises concerns regarding the consequences of a long-term VR exposure. Simultaneously, we need to address the individual differences and sensitivity of various users. For example, some users still suffer from cybersickness, which sense manipulation might contribute to or help to counter. It is crucial for both research and industry to determine what is causing these unpleasant symptoms and learn how to control them. Whether users will develop an increased tolerance to the factors causing cybersickness after long-term exposure to VR is still an unanswered question.

References

1. F. Steinicke et al., *Human Walking in Virtual Environments*, Springer, 2013.
2. S. Razzaque, "Redirected Walking," PhD dissertation, Univ. of North Carolina at Chapel Hill, 2005.
3. T. Field, S. Bay, and P. Vamplew, "Generalised Algorithms for Redirected Walking in Virtual Environments," *Proc. Int'l Conf. Artificial Intelligence in Science and Technology* (AISAT), 2004, pp. 58–63.
4. T.C. Peck, H. Fuchs, and M.C. Whitton, "Improved Redirection with Distractors: A Large-Scale-Real-Walking Locomotion Interface and Its Effect on Navigation in Virtual Environments," *Proc. IEEE Virtual Reality Conf.*, 2010, pp. 35–38.

5. F. Steinicke et al., “Estimation of Detection Thresholds for Redirected Walking Techniques,” *IEEE Trans. Visualization and Computer Graphics*, vol. 16, no. 1, 2010, pp. 17–27.
6. T. Grechkin et al., “Revisiting Detection Thresholds for Redirected Walking?: Combining Translation and Curvature Gains,” *Proc. Symp. Applied Perception*, 2016, pp. 113–120.
7. Q. Sun, L.-Y. Wei, and A. Kaufman, “Mapping Virtual and Physical Reality,” *ACM Trans. Graphics*, vol. 35, no. 4, 2016, article 64.
8. T.C. Peck, H. Fuchs, and M.C. Whitton, “An Evaluation of Navigational Ability Comparing Redirected Free Exploration with Distractors to Walking-in-Place and Joystick Locomotion Interfaces,” *Proc. IEEE Virtual Reality Conf.*, 2011, pp. 55–62.
9. D.J. Simons and R.A. Rensink, “Change Blindness: Past, Present, and Future,” *Trends Cognitive Sciences*, vol. 9, no. 1, 2005, pp. 16–20.
10. E.A. Suma et al., “Leveraging Change Blindness for Redirection in Virtual Environments,” *Proc. IEEE Virtual Reality Conf.*, 2011, pp. 159–166.
11. E.A. Suma, D.M. Krum, and M. Bolas, “Redirection on Mixed Reality Walking Surfaces,” *Proc. IEEE VR Workshop Perceptual Illusions in Virtual Environments*, 2011, pp. 33–35.
12. E.A. Suma et al., “Impossible Spaces: Maximizing Natural Walking in Virtual Environments with Self-Overlapping Architecture,” *IEEE Trans. Visualization and Computer Graphics*, vol. 18, no. 4, 2012, pp. 555–564.
13. K. Vasylevska and H. Kaufmann, “Influence of Path Complexity on Spatial Overlap Perception in Virtual Environments,” *Proc. 25th Int’l Conf. Artificial Reality and Telexistence and 20th Eurographics Symp. Virtual Environments* (ICAT-EGVE), 2015, pp. 159–166.
14. K. Vasylevska and H. Kaufmann, “Towards Efficient Spatial Compression in Self-Overlapping Virtual Environments,” *Proc. IEEE Symp. 3D User Interfaces* (3DU), 2017, pp. 12–21.
15. I. Moar and G.H. Bower, “Inconsistency in Spatial Knowledge,” *Memory and Cognition*, vol. 11, no. 2, 1983, pp. 107–113.
16. K. Vasylevska et al., “Flexible Spaces?: Dynamic Layout Generation for Infinite Walking in Virtual Environments,” *Proc. IEEE Symp. 3D User Interfaces* (3DU), 2013, pp. 39–42.
17. K. Vasylevska and H. Kaufmann, “Influence of Vertical Navigation Metaphors on Presence,” *Challenging Presence: Proc. 15th Int’l Conf. Presence*, A. Felnhofer and O.D. Kothgassner, eds., 2014, pp. 205–212.

Khrystyna Vasylevska *is a PhD candidate in the School of Informatics at TU Wien. Her research interests include immersive VR, redirected walking, 3D user interface design, brain-computer interfaces, and AI. Vasylevska has an MS in computer science from the National Technical University of Ukraine “Kyiv Polytechnic Institute.” Contact her at khrystyna.vasylevska@tuwien.ac.at.*

Hannes Kaufmann *is an associate professor in the Institute of Software Technology and Interactive Systems at TU Wien. His research interests include virtual and augmented reality, real-time ray-tracing for AR, wide-area tracking, motion capture, redirected walking and 3D user interfaces. Kaufmann has a PhD in technical sciences from TU Wien. Contact him at hannes.kaufmann@tuwien.ac.at.*

Contact department editors Frank Steinicke at frank.steinicke@uni-hamburg.de and Wolfgang Stuerzlinger at w.s@sfu.ca.

This article originally appeared in IEEE Computer Graphics and Applications, *vol. 37, no. 5, 2017.*

Computer Security as Civil Defense

Marilyn Wolf, Georgia Tech

Given the prevalence of computer systems, we must change our approaches by ensuring that civilians and companies can become responsible for much of their own cyberdefense.

It is time for us to treat computer and information security and safety as civil defense issues. I use the term *civil defense* here in its classic sense: the protection of civilians against military attack and natural disasters. Computer systems, both IT and cyberphysical (CPS), or the Internet of Things (IoT) can wreak widespread and long-lasting damage to civilian lives and property. Given the huge attack surface presented by civilian systems, we have no choice but to rely on civilians for a great deal of their own cyberdefense. Ensuring that civilians are prepared for cyberattacks and mishaps will require changes in our approaches to both technology and policy.

THE STAKES ARE HIGH

Let's keep in mind the huge stakes involved by reviewing a few recent incidents.

- The Target retail chain was the victim of a large data breach in 2013.[5] The attackers gained access to 11 GB of data. As a result, Target sent notices to 110 million credit and debit card holders.
- The Notpetya attack of 2017[3] targeted data and system configurations at several companies and resulted in extensive interruptions of company operations as well as a lengthy recovery process.
- A 2015 cyberphysical attack on Ukrainian electric power facilities resulted in a temporary loss of electrical service to more than 100,000 customers.[2]
- A cyberattack took down the computer systems of the Erie County Medical Center for six weeks in 2017.[1] Medical staff relied on paper documentation during the outage.

These serious examples of the damage that can be caused by computational attacks may, in fact, not provide us with a sufficiently bleak picture of worst-case damage. Reasonable people may be concerned that we could see much worse in the future at the hands of a capable and determined adversary.

Embedded computers are now in an astonishing variety of physical objects. Although computers have improved physical systems in many ways, these innovations also mean that we can no longer treat computer security and physical safety as separate topics. Safe and secure cyberphysical and IoT systems were the subjects of a special

Digital Object Identifier 10.1109/MC.2019.2891980
Date of publication: 11 March 2019

EDITOR DIMITRIOS SERPANOS
ISI/ATHENA and University of Patras; serpanos@computer.org

issue of *Proceedings of the IEEE* edited by Dimitrios Serpanos (this column's editor) and myself.[9] Security clearly affects safety; safety also influences our approach to computer security.

All nations need to be concerned about their cyber civil defense and readiness. Beyond nation-to-nation strife, nonstate actors could also carry out attacks, the effects of which give them a much broader reach. The 9/11 attacks showed that physical attacks with large effects can be planned and carried out by small groups;[6] we should be similarly concerned about the potential for large-scale computational attacks carried out by relatively small groups from well within their own safe havens.

CYBER AND CYPERPHYSICAL THREATS

Several types of threats are posed by cyber and cyberphysical attacks.

- Disruptions of service can affect both information systems and physical systems. The lines between IT and CPS/ the IoT are often blurry. As one example, IT failures at three U. S. airlines caused flight delays and cancelations.[4,7,8]
- Identity theft enables follow-on crimes. Beyond credit card fraud, attackers could use stolen credentials for improper access to facilities or data.
- Cyberphysical attacks can damage equipment. The Ukrainian power grid attack targeted power control devices but operated nondestructively, allowing workers to manually reset the equipment. A variation of the attack could have resulted in permanent damage. Industrial equipment often has replacement lead times measured in weeks or months, resulting in extended outages. A large-scale attack damaging an unusually large amount of equipment could further increase these backlogs, as could attacks on the facilities that manufacture such equipment.

DEFENDING AGAINST CIVIL CYBERTHREATS

Broadly speaking, we can identify several goals of computer civil defense: 1) protect the integrity of data, 2) protect the timely transfer of data, and 3) protect physical equipment. These goals are challenging in themselves. Computer civil defense is made even harder because of the wide variation in equipment and configuration and computer system operators' relative lack of expertise.

We can identify technical steps, ranging from known best practices to research topics, that can reduce cyberthreats. Some of these methods should be practiced by manufacturers. Root-of-trust design, which ensures that critical software can be traced back to a trusted source, is employed in practice but not universally. Root-of-trust design uses a combination of hardware and software methods: digital signatures for software are checked, access privileges for trusted versus nontrusted software are enforced, and digital signatures may be applied at several levels of deployment.

More controversial is a move toward lessened reliance on software updates. This is one example of physical safety influencing our approach to computer security. Updating controllers for physical systems is difficult for several reasons. Shutting down equipment for updates and then returning the equipment to operation depend not only on the computers but also on the equipment. Power-generating equipment may take several hours to come online. Chemical plants may require hours or days for a shutdown/restart cycle. Moreover, software for safety-critical systems is held to a high standard; fast updates to correct security-related bugs may not be possible while also ensuring that the updates do not cause further problems. We need software-engineering methods that result in fewer bug-fix distributions.

Design techniques for graceful degradation have received extensive attention over many decades. However, these methods are applied primarily in certain types of high-reliability systems.

Security clearly affects safety; safety also influences our approach to computer security.

Attacks that disrupt operations on IT systems suggest that more types of systems should be designed to provide some functionality in the face of failures to other parts of the system. Defense-in-depth methods are not consistently applied. The Target attack, for example, came through a cybersecurity weakness of a refrigeration contractor. System design should also take into account the time required to recover from attacks. The six weeks required to recover from the attack at the Erie County Medical Center is not an isolated example. Long recovery times amplify the damage caused by an attack.

Cyberphysical systems are sensitive not just to data values but also to timing—we can disrupt many control systems merely by delaying critical data without changing a single bit of information. Research has developed some architectures that preserve timing properties. *Timing resilience*—the detection of timing problems and responses to preserve system function—deserves more study. IT-oriented approaches to CPS and IoT security tend to treat these systems as collections of input–output devices to reduce the safety and security problems of traditional IT approaches. In fact, cyberphysical and IoT systems perform distributed real-time computations that require new security and safety methodologies.

Some quasi-technical factors also contribute to cybersecurity threats. Some IT personnel have received relatively little formal training in IT after promotion from technician or support roles. Training in cybersecurity is relatively new and may not have reached all practitioners. Personnel with training and experience in cyberphysical or IoT security are even harder to find.

Computer people pride themselves on the generality of computers. The result is that we see a huge variation in deployment configurations for devices and networks. Such variation makes security holes more likely and security properties harder to monitor. The use of more typical configurations for devices and networks would help to reduce problems and simplify fixes when problems are identified.

POLICIES TO RAISE AWARENESS

Policy will need to reinforce our understanding of risks and how we can best prepare ourselves. Organizations can help to educate the citizenry and encourage cybersecurity efforts. Such organizations will need to operate locally and provide a personal touch—ad campaigns won't cause enough people to change their ways. Organizations could be independent nonprofits, supported by local or state government, or national government organizations. We should expect that these organizations will cooperate to provide service—national or international organizations may provide expertise on specialized topics that smaller organizations cannot afford.

An important role of cybercivil defense organizations can be to disseminate useful information and provide training. The 21st century equivalents to pamphlets on bomb shelter construction could provide useful information to individuals and companies on how to prepare for cyberattacks. The cyber equivalent of duck-and-cover drills could educate citizens on the nature of threats and appropriate responses to unexpected events. Consider, for example, an attack on automobiles that interferes with their operation while on the road—a little preparation and practice could drivers them how to react to minimize risk.

Governments should consider encouraging or requiring reporting. Cyberattacks are not always reported by companies because of concerns about bad publicity or reliability. In contrast, accidents in several domains, such as transportation, are required by law to be reported. Information gleaned from attacks can be used to learn about attackers' methods and develop responses. Reporting systems can be designed to protect confidential data while providing useful public knowledge—patent litigation regularly uses protection orders for confidential data while conducting the main business of the case in public.

The U.S. National Transportation Board keeps public databases of aviation accidents (https://www.ntsb.gov/_layouts/ntsb.aviation/index.aspx) and railroad accidents (https://www.ntsb.gov/investigations/AccidentReports/Pages/railroad.aspx) that serve as examples of incident reporting and analysis. In some cases, safety recommendations or maintenance alerts may be made as a result of accidents.

In contrast, the National Vulnerability Database (NVD) maintained by the U.S. National Institute for Standards and Technology (https://nvd.nist.gov/) concentrates on code. The NVD defines *vulnerability* as "a weakness in the computational logic (e.g., code) found in software and hardware components that, when exploited, results in a negative impact to confidentiality, integrity, or availability" (https://nvd.nist.gov/vuln). A code-centric view of security minimizes the importance of system architecture and procedures followed by personnel.

A code-centric view of security minimizes the importance of system architecture and procedures followed by personnel.

Mandatory standards may be appropriate in some cases—for example, aircraft certification takes into account some aspects of cybersecurity and software safety. Mandates may help to overcome manufacturer inertia, with air bag regulations providing a classic example.

REGULATIONS AND STANDARDS

We need to be sure that export regulations do not unnecessarily restrict technologies that promote cybersecurity and safety. Some technologies will always be closely guarded. Regulators should take into account both risks and rewards when designing protections. Global supply chains mean that export controls have a broad reach that may keep important technologies from being adopted. Also, Internet attacks can be conducted by devices that have never entered the country.

Regulators need to treat cybersecurity and safety as top-of-the-list concerns. Electric power utilities put a great deal of effort into traditional reliability in case of storms and natural disasters; regulators require utilities to be prepared for such events and impose fines for certain types of power outages. Cyberthreats are arguably a lower priority at some utilities because their regulators do not place high importance on such threats. Cyberattacks have been much less frequent than, for example, weather-caused outages. Unfortunately, the consequences of a cyberattack could be huge and long lasting. Regulators need to find ways to encourage utilities of all types—for example, electric, natural gas, water, sewage, and transportation—to plan for these new threats.

Voluntary standards have proven useful in other domains. The Energy Star ratings used in the United States were created by the federal government and are voluntary. A wide range of consumer products advertise their Energy Star ratings. Manufacturers can employ voluntary systems to advertise their security capabilities and allow consumers to vote with their wallets.

Cyberthreats to our data and our physical world are real, and they will not go away. The pervasive adoption of computer technology has given us huge benefits but also new types of risk. A civil defense approach to cybersecurity and safety can help the citizenry protect itself against attacks and effectively respond to the inevitable attempts by bad actors to interfere with daily life. C

REFERENCES

1. CBS News, "Inside the New York hospital hackers took down for 6 weeks," Aug. 18, 2017. [Online]. Available: https://www.cbsnews.com/news/cbsn-on-assignment-hackers-targeting-medical-industry-hospitals/
2. D. Goodin, "First known hacker-caused power outage signals troubling escalation," *ArsTechnia*, Jan. 14, 2016. [Online]. Available: http://arstechnica.com/security/2016/01/first-known-hacker-caused-power-outage-signals-troubling-escalation/
3. A. Greenberg, "The untold story of Notpetya, the most devastating cyberattack in history," *Wired*, Aug. 22, 2018. [Online]. Available: https://www.wired.com/story/notpetya-cyberattack-ukraine-russia-code-crashed-the-world/
4. A. Halsey III, "Delta computers crash, causing delays and cancellations. Experts say it shouldn't have happened," *The Washington Post*, Aug. 8, 2016. [Online]. Available: https://www.washingtonpost.com/local/trafficandcommuting/delta-airlines-computer-systems-crash-causing-flight-delays-and-cancellations/2016/08/08/7d5e8fa0-5d72-11e6-af8e-54aa2e849447_story.html?utm_term=.cb218d77d206
5. M. Kassner, "Anatomy of the Target data breach: Missed opportunities and lessons learned," ZDNet, Feb. 2, 2015. [Online]. Available: https://www.zdnet.com/article/anatomy-of-the-target-data-breach-missed-opportunities-and-lessons-learned/
6. National Commission on Terrorist Attacks Upon the United States, "The 9/11 report," July 22, 2004. [Online]. Available: https://www.9-11commission.gov/report/
7. J. W. Moyer, D. Hedgpeth, and F. Siddiqui, "Southwest Airlines computer glitch causes cancellations, delays for third day," *The Washington Post*, July 22, 2016. [Online]. Available: https://www.washingtonpost.com/news/dr-gridlock/wp/2016/07/21/long-lines-for-southwest-airlines-passengers-at-area-airports/
8. E. Ortiz, J. Shamlian, and T. Costello, "United Airlines flights no longer grounded, delays remain," NBC News, July 8, 2015. [Online]. Available: http://www.nbcnews.com/business/travel/united-airlines-passengers-say-flights-grounded-nation-wide-n388536
9. M. Wolf and D. Serpanos, "Safety and *Security in Cyber-Physical Systems* and Internet-of-Things systems," *Proc. IEEE*, vol. 106, no. 1, pp. 9–20, Jan. 2018. doi: 10.1109/JPROC.2017.2781198.

MARILYN WOLF is the Rhesa "Ray" S. Farmer, Jr., distinguished chair in Embedded Computing Systems and Georgia Research Alliance Eminent Scholar at the Georgia Institute of Technology. She is a Fellow of the IEEE and ACM. Contact her at wolf@ece.gatech.edu.

This article originally appeared in Computer, *vol. 52, no. 1, 2019.*

EDITOR DIMITRIOS SERPANOS
ISI/ATHENA and University of Patras;
serpanos@computer.org

Cyber-Physical Systems and Digital Twins in the Industrial Internet of Things

Christos Koulamas and Athanasios Kalogeras,
Industrial Systems Institute/ATHENA

A digital twin is a virtual representation that serves as the real-time digital counterpart of a physical object or process and addresses every instance for its total life cycle.

Operational models and other virtual representations of cyber-physical systems (CPSs) are a common industrial engineering practice today. The evolution of Internet of Things (IoT) and AI technologies enables complex interactions of such virtual representations for the total lifetime of system instances under the digital twin (DT) concept, which poses a number of challenges for its seamless integration in the modern industrial environment.

CPSs lie at the cross section of the physical and digital worlds. Integrating physical processes and computer systems is the main challenge presented by them, as the computational cyber part continuously senses the state of the physical system and applies decisions and actions for its control. CPSs present a wide range of applications in different sectors, including manufacturing, energy, health care, consumer services, and monitoring of critical infrastructures.

CPSs are mostly networked systems characterized by distribution of functions and often wireless connectivity between intelligent physical devices, providing sensing and actuating as well as control capabilities. Real-time behavior is a critical challenge, as the continuous monitoring and control of the physical world has to be ascertained. They represent complex, flexible, and adaptive systems, whose constituent elements are characterized by increased autonomy and intelligence. Cybersecurity mechanisms need to be integrated in a holistic approach

Digital Object Identifier 10.1109/MC.2018.2876181
Date of publication: 15 January 2019

toward detection of attacks, resilience, and privacy concerns.

The Industrial IoT (IIoT) is an enabling technology of CPSs, providing the networking infrastructure for physical objects to sense, communicate, and interact. The spread of the IIoT has led to an explosion of data and information. With 21 billion connected things by 2020,[1] the IIoT market is estimated to add $14.2 trillion to the global economy by 2030.[2] Manufacturing, connected logistics and transportation, and energy and utilities represent the three largest markets for the IIoT. Digitization of these markets creates a plethora of data and new opportunities for companies to extract knowledge out of these data.

THE DT CONCEPT

The increasing availability and ubiquity of real-time operational data, as well as the boost of AI implementation capabilities in learning and reasoning, represent drivers toward realizing a vision of physical products or processes having accompanying virtual representations that evolve throughout their entire life cycle. Such virtual representations, or DTs, represent real-time digital counterparts of physical objects. There is not any unique, globally accepted, and common closed definition of the DT concept; however, there are certain aspects on which most existing definitions agree.[7] A DT is virtual (that is, digital), it includes both static (that is, design documents, process specifications, and so forth) and dynamic (that is, data acquisition and simulation) parts, and it addresses every instance of its twin product or process for its total life cycle.

Could DT be a new marketing buzzword for a concept that already exists? There might be some truth in this: a part of the DTs' expected capabilities (for example, precise simulation of the physical thing's behavior) could already be utilized in current engineering practices. Still, the DT concept attempts to materialize a bidirectional integration of the digital and physical worlds, interconnecting physical things with their digital counterparts while also bringing to the physical world changes to its DT.

Gartner included DT as top strategic technology trend number four for 2018 (among the top 10 trends that will contribute to the intelligent digital mesh, that is, the integration of things, services, content, and people).

One direct utilization of a DT is in the field of asset management. Being the cyber twin of a physical thing and having access to real-time information regarding the physical thing as well as to related historical data, the DT can help optimize physical asset performance through efficient predictive and preventive maintenance operations, thus reducing overall maintenance costs and downtime.

Furthermore, the DT can simulate the behavior of the physical thing that it is twinned with—or of an associated process—and can thus contribute significantly to performance optimization. It can act as a tool for predictive analysis, predicting the performance of the physical thing or its associated process. If this example is enlarged in the scope of a digital enterprise, then the overall process, production system, or product may be optimized. Potential benefits include, among others, optimizing production scheduling, identifying potential bottlenecks, assessing asset utilization, and minimizing production lead times.

The DT offers a total life cycle approach with reference to its physical twin, either a thing or a process. Product design, new product launch, manufacturing process setup, and integrated supply chain management are facilitated. It is essential, though, to point out the principal difference of the virtual representation of a DT compared with well-known and widely used relevant modeling engineering in design, simulation, and testing: the permanent connection between the real and the virtual part for the total life cycle of a specific system instance. This connection means that information exchange between the system instance and its DT counterpart (that is, sensing and often also actuating infrastructure) can be part of a cyber-physical system on its own, depending on the type of information exchange—whether a real-time data flow or some systematic data collection that is integrated offline in the behavior of the DT.

INTEGRATION CHALLENGES

The need for a bidirectional life cycle-extended integration between the physical world and its DT mandates a relevant supporting reference architecture. Different initiatives deal with the IIoT providing relevant reference architectures, the most important of which are Industry 4.0, the Industrial Internet Consortium (IIC), and Society 5.0.

The Reference Architecture Model for Industry 4.0 (RAMI 4.0) is a three-dimensional model along three axes (hierarchy, architecture, and product life cycle), with IT security and data privacy as enablers (see the right side of Figure 1).[3] The Industrial Internet Reference Architecture (IIRA) of the IIC[4] comprises four different viewpoints: business, usage, functional, and implementation. The functional viewpoint (see the left side of Figure 1) is, in turn, divided into five domains (control, operations, information, application, and business), four cross-cutting functions (connectivity, distributed data management, analytics, and intelligent and resilient control), and six system characteristics (safety, security, resilience, reliability, privacy, and scalability). The implementation viewpoint utilizes a three-tier implementation architecture comprising the enterprise, platform, and edge tiers (see the lower part of Figure 1). A mapping between RAMI 4.0 and IIRA is also possible.[5] Finally, Society 5.0 represents the related Japanese initiative driving toward a new hypersmart society and extending to different application domains. Its enabling technologies comprise the Internet of Things, big data, ambient intelligence, and robotics.[6]

The interweaving of the DT with its physical counterpart starts with the capture of data generated by sensing things in the manufacturing environment. This provides the DT with real-time data of the physical world it is twinned with. This step can be mapped to the asset layer of the architecture axis of RAMI 4.0. These data are then aggregated and combined with historical data pertaining to the manufacturing process as well as relevant data at the enterprise level. This step drives from the physical to the cyber part and corresponds to the integration layer of RAMI 4.0.

Then, at the communication layer, data move to the fog, the edge, or the cloud depending on the architecture that is followed. Data analytics are applied to these data, and useful information is derived. This information can guide some optimization in the product or process. Some action is triggered in the real world by the DT to achieve this.

Furthermore, as the cyber and physical perspectives of the DT may already blur the borders between the DT of a complex cyber-physical system and the system itself, embedded technology evolution has already started to challenge typical architectural patterns in the realization of DTs, which usually call for relatively heavy centralized computing power and relevant data center and cloud infrastructures. There are applications that have inherent characteristics and requirements that provide a natural fit to highly distributed intelligence at the edge. However, the benefits of similar setups, mainly network bandwidth and cloud processing cost reductions but also responsiveness, dependability, scalability, and security improvements, can be exploited in a wider set of domains, especially in fault detection and diagnosis for preventive and predictive maintenance.

There is ongoing research on enabling AI capabilities in embedded devices,[8,9] while specialized hardware and real-time embedded analytics frameworks are already in the market, justifying their necessity in various industrial settings.[10] This is expected to lead to a wider adoption of this specific architectural paradigm, considering the high achievable degree of containment for a relatively small "twin" of a tiny but critical "thing," which is able to create, train, consult, and adapt its DT onboard and in real time without any interaction with the cloud. Such twins can be then envisaged as capable of being combined in a system-of-systems fashion to create larger distributed models for DTs of highly complex cyber-physical systems.

The DT concept in the IIoT context generates a number of challenges. Its constituent elements are quite diverse. Product and production process models must be seamlessly integrated with simulation and prediction models, as well as with tools and systems dealing with data analytics and optimization; nontypical, embedded, and mobile computing platforms must also be considered. This interoperability challenge of combining completely different models, systems, and tools represents an area in need of significant research. Mapping and integration of the DT and its

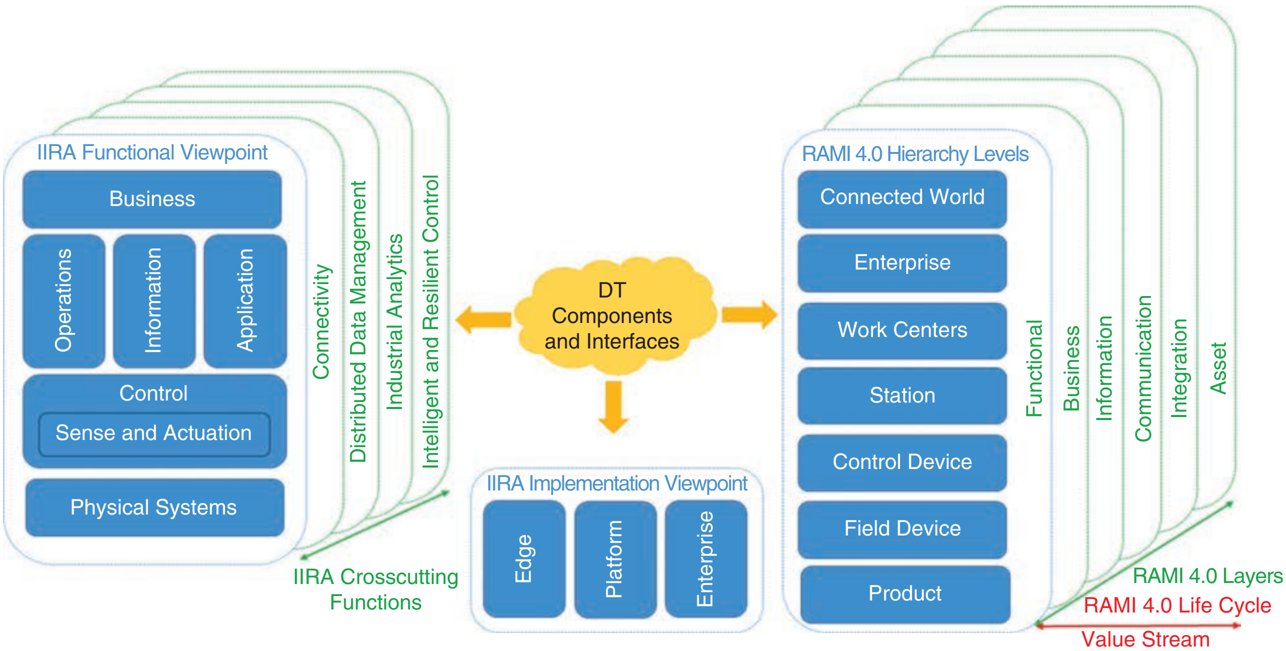


FIGURE 1. The CPSs and digital twins (DTs) in the Industrial IoT.

This article originally appeared in Computer, *vol. 51, no. 11, 2018.*

functionalities on the prevalent reference architectures for the IIoT is also a necessity (see Figure 1).

The DT is still mostly at a conceptual stage, in terms of demonstrating wide industrial adoption and becoming a well-defined engineering practice within the industry. There is a need for research dealing with the previously listed challenges. Further, there is a need for a unified framework to build, out of the corresponding physical world model, its DT. This framework should offer the full range of tools necessary for DT operation.

REFERENCES

1. M. Hung. (2017). Leading the IoT. Gartner insights on how to lead in a connected world. *Gartner*. [Online]. Available: https://www.gartner.com/imagesrv/books/iot/iotEbook_digital.pdf
2. Accenture. (2015). Winning with the industrial Internet of things. *Accenture*. [Online]. Available: https://www.accenture.com/t20160909T042713Z__w__/us-en/_acnmedia/Accenture/Conversion-Assets/DotCom/Documents/Global/PDF/Dualpub_11/Accenture-Industrial-Internet-of-Things-Positioning-Paper-Report-2015.pdfla=en
3. P. Adolfs, S. Berlik, W. Dorst, J. Friedrich, C. Gericke, M. Hankel, R. Heidel, M. Hoffmeister, C. Mosch, R. Pichler, U. Rauschecker, T. Schulz, K. Schweichhart, E. J. Steffens, M. Taube, I. Weber, M. Wollschlaeger, and S. Mätzler, "Reference Architecture Model Industrie 4.0 (RAMI4.0)," Deutsche Institut für Normung e.V. (DIN), Berlin, Germany, DIN SPEC 91345, 2016.
4. S.-W. Lin, B. Miller, J. Durand, G. Bleakley, A. Chigani, R. Martin, B. Murphy, and M. Crawford. (2017). The industrial Internet of things, volume G1: Reference architecture. *Industrial Internet Consortium*. [Online]. Available: https://www.iiconsortium.org/IIC_PUB_G1_V1.80_2017-01-31.pdf
5. S.-W. Lin, B. Murphy, E. Clauer, U. Loewen, R. Neubert, G. Bachmann, M. Pai, and M. Hankel. (2017). Architecture alignment and interoperability Industrial Internet Consortium and Platform Industrie 4.0 Joint Whitepaper. *Industrial Internet Consortium*, Needham, MA. [Online]. Available: https://www.iiconsortium.org/pdf/JTG2_Whitepaper_final_20171205.pdf
6. Japanese Cabinet Office, "Society 5.0,'" *Japanese Cabinet Office*. Accessed on: Nov. 19, 2018. [Online]. Available: http://www8.cao.go.jp/cstp/english/society5_0/index.html
7. I. Negri, L. Fumagalli, and M. Macchi, "A review of the roles of digital twin in CPS-based production systems," *Procedia Manufacturing*, vol. 11, pp. 939–948, Sept. 2017.
8. N. D. Lane and P. Warden, "The deep (learning) transformation of mobile and embedded computing," *IEEE Computer*, vol. 51, no. 5, pp. 13–16, May 2018.
9. J. Tang, D. Sun, S. Liu, and J.-L. Gaudiot, "Enabling deep learning on IoT devices," *IEEE Computer*, vol. 50, no. 10, pp. 92–96, Oct. 2017.
10. C. Middleton. (2018) Digital twin AIs designed to learn at the edge. *Control Engineering*. [Online]. Available: https://www.controleng.com/single-article/digital-twin-ais-designed-to-learn-at-the-edge

CHRISTOS KOULAMAS is a research director at the Industrial Systems Institute/ATHENA Research Center. He is a Senior Member of IEEE. Contact him at koulamas@isi.gr.

ATHANASIOS KALOGERAS is a research director at the Industrial Systems Institute/ATHENA Research Center. He is a Senior Member of IEEE. Contact him at kalogeras@isi.gr.

IEEE COMPUTER SOCIETY

Harry H. Goode Memorial Award

Call for Award Nominations

Deadline: 1 October 2019

2019 Harry H. Goode Memorial Award Recipient

Marilyn C. Wolf

Georgia Institute of Technology

For contributions to embedded, hardware-software codesign, and real-time computer vision systems.

Named for a pioneer and leader in the field of systems engineering, the **Harry H. Goode Memorial Award** was established to encourage further developments in and honor outstanding contributions to the field of information processing sciences. A bronze medal and $2,000 are awarded by the IEEE Computer Society on the basis of achievements in the information processing field.

About Harry H. Goode

One of the first scientists to fully comprehend the powers and abilities of computers, Harry H. Goode formulated many principles of systems engineering and developed techniques for the design, analysis, and evaluation of large-scale systems. He was instrumental in initiating early systems projects, including the Typhoon computer and Whirlwind computer at MIT. He participated in the study that led to the creation of the Bomarc missile and conceived and developed the Air Defense Integrated System Project.

Nomination Requirements

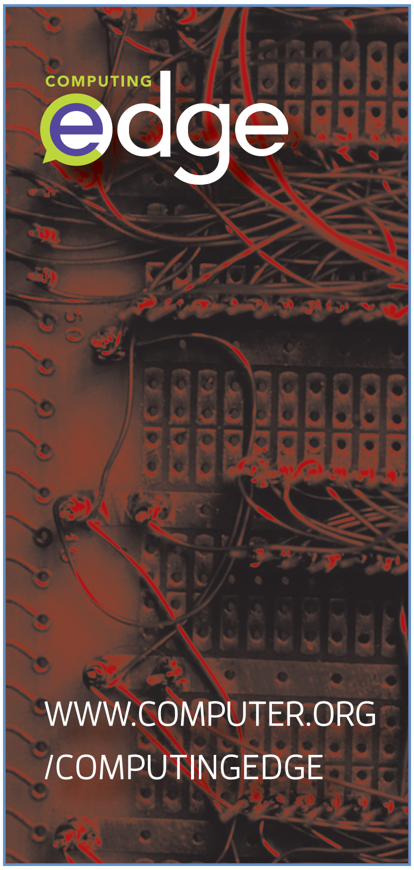
This award requires a minimum of three endorsements.

Nominations are being accepted electronically by **1 October 2019** to **bit.ly/harry-goode.**

Questions?

Visit **bit.ly/harry-goode**
or contact **awards@computer.org**

IEEE



UNIVERSITY OF WISCONSIN
PLATTEVILLE
Opportunity from the start

The College of Engineering, Mathematics and Science (EMS)

Chair of the Computer Science and Software Engineering (CSSE) Department

University of Wisconsin-Platteville is seeking a visionary leader to serve as the Department Chair of Computer Science and Software Engineering. We are pursuing exceptional candidates with proven leadership experience, demonstrated commitment to excellence in undergraduate education, and a strong track record of industry, federal, or state-funded research. The CSSE department is student-focused and is known for its hands-on, experiential approach and *highly sought-after graduates*.

This is a tenured faculty position with a start date of August 1, 2020.

The CSSE Department at UW-Platteville is growing and transforming, with several new tenure-track faculty, increasing student enrollments and research activities, and involvement in several campus initiatives. CSSE currently has 13 faculty and staff, serving approximately 400 students in three on-campus majors and one collaborative online degree. The Software Engineering program is one of two ABET-accredited SE programs in the state of Wisconsin.

For more information about our campus, the complete position description and how to apply, please visit our Career Center at **careers.uwplatt.edu**

IEEE COMPUTER SOCIETY

Richard E. Merwin Award for Distinguished Service

Call for Award Nominations
Deadline: 1 October 2019

The Richard E. Merwin Award is the highest-level volunteer service award of the IEEE Computer Society for outstanding service to the profession at large, including significant service to the IEEE Computer Society or its predecessor organizations.

About Richard Merwin

Richard Merwin was a pioneer in digital computer engineering who participated in the development of the ENIAC, MANIAC, and STRETCH computers. Despite a busy and productive technical career, Merwin found time to be active in professional societies, including the IEEE Computer Society, ACM, and AFIPS. His generosity of spirit and genuine helpfulness was an important element in the progress of the computer profession.

Award

A bronze medal and $5,000 honorarium are awarded.

Presentation

The Richard E. Merwin Award is presented at the IEEE Computer Society's Annual Awards Ceremony.

Nomination Requirements

This award requires 3 endorsements.

Nominations are being accepted electronically by **1 October 2019** to **bit.ly/richard-merwin**

Questions?

Email **awards@computer.org**

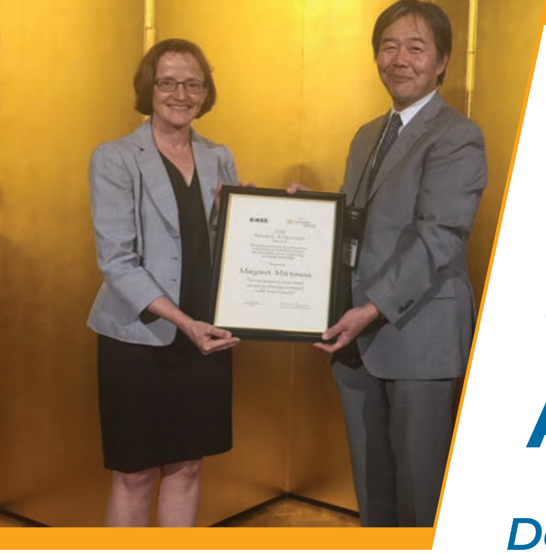
2018 Richard E. Merwin Award Recipient for Distinguished Service

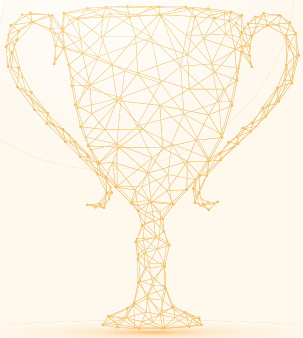
Sorel Reisman

California State University, Fullerton

For sustained contributions, leadership, and service to the IEEE Computer Society, IEEE, and the computing profession at large.

IEEE





Conference Calendar

Questions? Contact conferences@computer.org

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.

Find a region:	Africa ■	Australia ◆	North America ◗
	Asia ▲	Europe ●	South America ★

OCTOBER

1 October

- MCSoC (IEEE 13th Int'l Symposium on Embedded Multicore/Many-core Systems-on-Chip) ▲

2 October

- DFT (IEEE Int'l Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems) ●

12 October

- MICRO (52nd Annual IEEE/ACM Int'l Symposium on Microarchitecture) ◗

14 October

- ISMAR (IEEE Int'l Symposium on Mixed and Augmented Reality) ▲
- LCN (IEEE 44th Conf. on Local Computer Networks) ●
- VL/HCC (IEEE Symposium on Visual Languages and Human-Centric Computing) ◗

15 October

- AIPR (IEEE Applied Imagery Pattern Recognition Workshop) ◗

16 October

- FIE (IEEE Frontiers in Education Conf.) ◗

20 October

- VIS (IEEE Visualization Conf.) ◗

27 October

- ICCV (IEEE/CVF Int'l Conf. on Computer Vision) ▲

28 October

- EDOC (IEEE 23rd Int'l Enterprise Distributed Object Computing Conf.) ●
- ISSRE (IEEE 30th Int'l Symposium on Software Reliability Eng.) ●

NOVEMBER

4 November

- ICTAI (IEEE 31st Int'l Conf. on Tools with Artificial Intelligence) ◗

7 November

- SEC (IEEE/ACM Symposium on Edge Computing) ◗

8 November

- ICDM (IEEE Int'l Conf. on Data Mining) ▲

9 November

- FOCS (IEEE 60th Annual Symposium on Foundations of Computer Science) ◗

11 November

- ASE (34th IEEE/ACM Int'l Conf. on Automated Software Eng.) ◗

17 November

- ICCD (IEEE 37th Int'l Conf. on Computer Design) ▲
- SC19 (SC19: Int'l Conf. for High Performance Computing, Networking, Storage and Analysis) ◗

18 November

- BIBM (IEEE Int'l Conf. on Bioinformatics and Biomedicine)

DECEMBER

3 December

- RTSS (IEEE Real-Time Systems Symposium) ▲

4 December

- IREHI (IEEE Int'l Rural and Elderly Health Informatics Conf.) ■

9 December

- AIVR (IEEE Int'l Conf. on Artificial Intelligence and Virtual Reality)
- Big Data (IEEE Int'l Conf. on Big Data)
- ISM (IEEE Int'l Symposium on Multimedia)

10 December

- ISSPIT (IEEE Int'l Symposium on Signal Processing and Information Technology) ▲

16 December

- CDKE (IEEE Int'l Conf. on Conversational Data & Knowledge Eng.)

2020

January

13 January

- ICCPS (Int'l Conf. on Cyber-Physical Systems) ●

February

3 February

- ICSC (IEEE 14th Int'l Conf. on Semantic Computing)

18 February

- SANER (IEEE 27th Int'l Conf. on Software Analysis, Evolution and Reengineering)

19 February

- BigComp (IEEE Int'l Conf. on Big Data and Smart Computing) ▲

22 February

- CGO (IEEE/ACM Int'l Symposium on Code Generation and Optimization)

March

2 March

- WACV (IEEE Winter Conf. on Applications of Computer Vision)

9 March

- DATE (Design, Automation & Test in Europe Conf. & Exhibition) ●
- IRC (4th IEEE Int'l Conf. on Robotic Computing) ▲

16 March

- ICSA (IEEE Int'l Conf. on Software Architecture) ★

22 March

- VR (IEEE Conf. on Virtual Reality and 3D User Interfaces)

23 March

- ICST (13th IEEE Conf. on Software Testing, Validation and Verification) ●
- PerCom (IEEE Int'l Conf. on Pervasive Computing and Communications)

April

5 April

- ISPASS (Int'l Symposium on Performance Analysis of Systems and Software)

14 April

- PacificVis (IEEE Pacific Visualization Symposium) (Location TBD)

20 April

- ICDE (IEEE 36th Int'l Conf. on Data Eng.)

May

3 May

- FCCM (IEEE 28th Annual Int'l Symposium on Field-Programmable Custom Computing Machines)

IPDPS 2020 New Orleans

New Orleans 18 - 22 May

ipdps.org

34th IEEE International Parallel and Distributed Processing Symposium

IPDPS 2020 CALL FOR PAPERS

The five-day IPDPS program includes three days of contributed papers, invited speakers, industry participation, and student programs, framed by two days of workshops with peer reviewed papers that complement and broaden the main program. For full details, see www.ipdps.org.

Authors for the main conference are invited to submit manuscripts that present original unpublished research in all areas of parallel and distributed processing, including the development of experimental or commercial systems. Work focusing on emerging technologies and interdisciplinary work covering multiple IPDPS areas are especially welcome. Topics of interest include:

- **Parallel and distributed computing theory and algorithms (Algorithms)**
- **Experiments and practice in parallel and distributed computing (Experiments)**
- **Programming models, compilers and runtimes for parallel applications and systems (Programming Models & Compilers)**
- **System software and middleware for parallel and distributed systems (System Software)**
- **Architecture**
- **Multidisciplinary**

• Abstracts due	October 7, 2019
• Submissions due	October 14, 2019
• Preliminary decisions	December 9, 2019
• Final submissions due	January 6, 2020
• Final notification	January 20, 2020

Sponsored by

IEEE COMPUTER SOCIETY TCPP

Technical Committee on Parallel Processing

acm In-Cooperation

ACM SIGARCH

sighpc

GENERAL CO-CHAIRS

Anu Bourgeois *(Georgia State University, USA)*
Ramachandran Vaidyanathan *(Louisiana State University, USA)*

PROGRAM CHAIR

Yuanyuan Yang *(NSF and Stony Brook University, USA)*

PROGRAM AREA CHAIRS AND VICE CHAIRS

- **Algorithms:**
  Xiaotie Deng *(Peking University, China)* and
  Songtao Guo *(Chongqing University, China)*
- **Architecture:**
  Ahmed Louri *(George Washington University, USA)* and
  Avinash Karanth *(Ohio University, USA)*
- **Experiments:**
  Xin Yuan *(Florida State University, USA)* and
  Scott Pakin *(Los Alamos National Laboratory, USA)*
- **System Software:**
  Alan Sussman *(NSF and University of Maryland, College Park, USA)* and
  Zhiling Lan *(Illinois Institute of Technology, USA)*
- **Programming Models & Compilers:**
  Rudolf Eigenmann *(University of Delaware, USA)*
  Zhiyuan Li *(Purdue University, USA)*
- **Multidisciplinary:**
  Manish Parashar *(NSF and Rutgers University, USA)*
  Ivona Brandić *(Vienna University of Technology, Austria)*

New Orleans *is one of the most eccentric and lively cities in the world. Whatever your interests are, New Orleans has you covered. From its diverse culture, distinctive cuisine, rich history, colorful celebrations, live music, vibrant nightlife, and world-class restaurants, there is something for everyone. It is home to a number of engaging museums, including the World War II Museum, the New Orleans Museum of Art, the Historic Voodoo Museum, Mardi Gras World, and the Pharmacy Museum. A visit wouldn't be complete without a swamp tour, Mississippi river cruise and a stop at Café du Monde for beignets. Join IPDPS at the Hilton New Orleans Riverside in 2020 to find out what makes New Orleans so unique and special.*