COMPUTING

edge

ADVANCING THE INTERNET OF THINGS

Also in this issue:

> Turning Back the Clock

> IT Project Failures: What Management Can Learn

APRIL 2016

www.computer.org

IEEE

IEEE computer society

CELEBRATING 70 YEARS

ISADS 2017

CALL FOR PAPERS

The 13th International Symposium on Autonomous Decentralized Systems (ISADS)

March 22 - 24, 2017 Bangkok, Thailand

http://ieeethailand.org/isads2017.php

Sponsored by: The Institute of Electrical and Electronics Engineers; IEEE Computer Society

In Cooperation with: IEICE Communications Society; IEICE Information and Systems Society; International Federation for Information Processing; International Federation of Automatic Control; Object Management Group

Supported by: Chulalongkorn University (CU); Thai-Nichi Institute of Technology (TNI); Thailand Convention and Exhibition Bureau (TCEB)

Honorary Chairs
Bundhit Eua-arporn, Chulalongkorn Univ. Thailand
Bundhit Rojarayanon, TNI, Thailand
Katsuhiko Shirai, Open University of Japan, Japan

General Chair
Jitkasem Ngamnil. Mahanakorn Univ., Thailand

Program Co-Chairs
Thavatchai Tayjasanant, Chulalongkorn Univ., Thailand
Yinong Chen, Arizona State Univ., USA
Xiaodong Lu, ENRI, Japan
Leo van Moergestel, Hogeschool Utrecht, Netherlands

Program Committee
Farokh B. Bastani, University of Texas-Dallas, USA
Renée Bryce, University of North Texas, USA
Ing-Ray Chen, Virginia Polytechnic Institute and State Univ., USA
Bruce C. Fan, Google, USA
Yan Luo, University of Massachusetts Lowell, USA
Khalid Mahmood, Oakland University, USA
Carlos Perez, Bank of Mexico, Mexico,
Wei-Tek Tsai, Arizona State University, USA
I-Ling Yen, University of Texas-Dallas
Weider Yu, San José State University, USA
Robbert van Renesse, Cornell University USA
Alfonso Fuggeta, CEFRIEL, Italy
John-Jules Meyer, Utrecht University, The Netherlands
Erik Puik, Utrecht University of Applied Sciences, The Netherlands
Ingo Schwab, Karlsruhe University of Applied Sciences, Germany
Egons Lavendelis, Riga Technical University, Latvia
Vadim Ermolayev, Zaporozhye National University, Ukraine
Michael Klafft, Jade University of Applied Sciences, Germany
Minna Lanz, Tampere University of Technology, Finland
Cees Witteveen, Delft Univ. of Tech., The Netherlands
Jie Xu, Leeds University, UK
Helen Karatza, Aristotle University of Thessaloniki, Greece
Masaki Aida, Tokyo Metropolitan University, Japan
Chisa Takano, Hiroshima City University, Japan
Atsushi Ito, Utsunomiya University, Japan
Hiroaki Morino, Shibaura Institute of Technology, Japan
Tadashi Koga, Electronic Navigation Research Institute, Japan
Takashi Kunifuji, East Japan Railway Company, Japan
Yuichi Yagawa, Hitachi, Japan
Xiaoying Bai,, Tsinghua University, China
Yingxu Lai, Beijing University of Technology, China
Qingguo Zhou, Lanzhou University, China
Fan Wei, Xi'an Shiyou University, China
Anthony Y. H. Liao, Asia Univ., Taiwan
NaKorn Chantasorn, NSTDA, Thailand
Pramual Suteecharuwat, Chulalongkorn Univ., Thailand
Pornphisud Mongkhonvanit, Siam Technological College, Thailand
Kamol Kaemarungsi, NECTEC, Thailand
Suvit Nakpeerayuth, Chulalongkorn Univ., Thailand
Warakorn Srichavengsup, TNI, Thailand

Steering Committee
Kinji Mori, Waseda Univ., Japan (Chair)
Farokh B. Bastani, Univ. of Texas-Dallas, USA
Yinong Chen, Arizona State Univ., USA
Masaki Ogata, JR East, Japan
Radu Popescu-Zeletin, Technical Univ. Berlin, Germany
Richard Mark Soley, OMG, USA
Jeffrey Tsai, Asia University, Taiwan

Local Arrangement Chair
Vidhya Rugpong, EGAT, Thailand

Finance Chair
Kanchana Miskate, MEA, Thailand

Publicity / Registration Committee Chair
Rujipan Sampanna, Bangkok Univ., Thailand

Publication Committee Chair
Veerapon Vateekul, Chulalongkorn Univ., Thailan

Secretary Generals
Yuri Kawakami, Waseda Univ., Japan
Titichaya Thanamitsomboon, TNI, Thailand

Scope

Opportunities and challenges for implementing highly complex, efficient, and dependable business and control systems have been steadily increasing, driven by the continuous growth in the power, intelligence, adaptiveness and openness of technologies and standards applied in computing, communication and control systems. Dynamically changing social and economic situations demand the next-generation of systems to be based on adaptive, reusable, and internet and Web-enabled technologies and applications. Such systems are expected to have the characteristics of living systems composed of largely autonomous and decentralized components. Such systems are called Autonomous Decentralized Systems (ADS). The International Symposium on Autonomous Decentralized Systems (ISADS) has been the premier events in the past twenty-six years to have successfully addressed these challenges. The 13th ISADS 2017 will continue to focus on the advancements and innovations in ADS concepts, technologies, applications strategic issues, and other related topics. The special topic for ISADS 2017 is "Technology and Business Innovation through Structure Change of Society and Life". We invite research papers, workshop proposals, panel proposals, and student papers on, but not limited to, the following topics:

- Internet of Things, cyber-physical systems, reactive systems, distributed embedded systems
- Industry 4.0, and autonomous robotics and transportation
- Ad-hoc networks, sensor networks, advanced network infrastructures and internetworking
- Architecture-driven and model-driven development of distributed applications
- Assurance, fault tolerance, on-line expansion, on-line-maintenance, and resilience
- Autonomous and decentralized services, including service architecture, protocols, and collaboration
- Cloud computing and big data analysis
- Distributed and collaborative development, test and maintenance, and development infrastructure of high-quality software systems, cloud computing, and service-oriented architecture
- Heterogeneous distributed information / control systems, mobile agent / computer-supported cooperative works
- Modeling and simulation of autonomous services and service-oriented application composition
- Network and system security and safety
- Autonomous Decentralized System Applications in smart grids
- Autonomous Decentralized System Applications in railway engineering
- Novel applications, including e-business, e-commerce and e-government; telecommunications; information service systems; manufacturing systems; real-time event management; office automation; traffic and transportation control; supply chains; environmental/emergency protection; networked health and medical systems; intelligent home control; embedded systems for automotive and avionics applications

Information for Authors

Papers should describe original work and be of 6 to 8 pages in IEEE double-column conference paper format. Workshop paper should be of length 5 to 6 pages. Papers should include: title, authors, affiliations, 150-word abstract and list of keywords. Please identify the contact author clearly, including name, position, mailing address, telephone and fax numbers, and email address. At least one of the authors of each accepted paper must register and present the paper at ISADS 2017. Authors must submit their manuscripts electronically following the instructions at the ISADS 2017 web site at: http://ieeethailand.org/isads2017.php

Information for Workshop and Panel Organizers

Workshop and panel proposals should include: title, organizer affiliation, position, mailing address, telephone and fax numbers, email address, and a draft call-for-papers, including Chairs, committees and submission deadline. Workshop and Panel proposals must be e-mailed to the program chair, Dr. Thavatchai Tayjasanant (thavatchai.t@chula.ac.th)

Important Dates

June 15, 2016: Workshop proposals are due.
August 31, 2016: All papers and panel proposals are due.
November 15, 2016: Authors and panel organizers notified of acceptance.
December 31, 2016: Camera-ready copies of accepted papers and panelist position papers.

General Information

For general information, visit the ISADS 2017 web site at: http://ieeethailand.org/isads2017.php or email the program chair, Dr. Thavatchai Tayjasanant (thavatchai.t@chula.ac.th or taytaycu@gmail.com)

COMPUTING edge

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

Printed with inks containing soy and/or vegetable oils

IEEE

IEEE computer society
CELEBRATING 70 YEARS

BPA WORLDWIDE

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

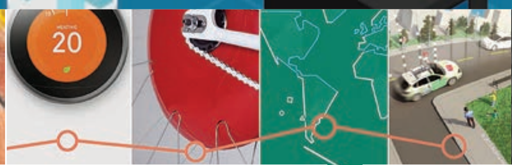
Postmaster: Send address changes to ComputingEdge-IEEE Membership Processing Dept., 445 Hoes Lane, Piscataway, NJ 08855. Periodicals Postage Paid at New York, New York, and at additional mailing offices. Printed in USA.

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


HISCOX
business insurance®
"My business runs on big ideas. And a tiny budget."
Invest more in your company's success, and less in its business insurance. Get a customized policy starting as low as $22.50/mo.
Get a fast, free quote at Hiscox.com/bigideas or call our licensed insurance agents at 866-507-0461 Mon–Fri, 8:00am–10:00pm ET.
#encouragecourage.
© 2016 Hiscox Inc. All rights reserved.

APRIL 2016 • VOLUME 2, NUMBER 4

COMPUTING edge



52

IT Project Failures: What Management Can Learn

Departments

Using 2D Maps for 3D Localization

Leila De Floriani, University of Genoa

This installment highlighting the work published in IEEE Computer Society journals comes from IEEE Transactions on Visualization and Computer Graphics.

Today's smartphones and tablets allow augmented reality at consumer prices by superimposing information registered onto the device's live video preview. However, despite augmented reality's promise—and attention from Google, Microsoft, and other major technology providers—it's still not widely used.

One obstacle to widespread adoption is the lack of outdoor localization technology that can operate with reasonable resources. The mobile devices themselves are inexpensive. However, outdoor localization that scales to large areas, such as an entire city, typically requires access to costly infrastructures. First, a detailed 3D reconstruction of the environment must be captured and stored in a database. Second, a computationally expensive image-matching process from the live video image to the database must be run to determine the device's location. Computation can be pushed to the cloud, but not without adding network latency that hurts real-time processing.

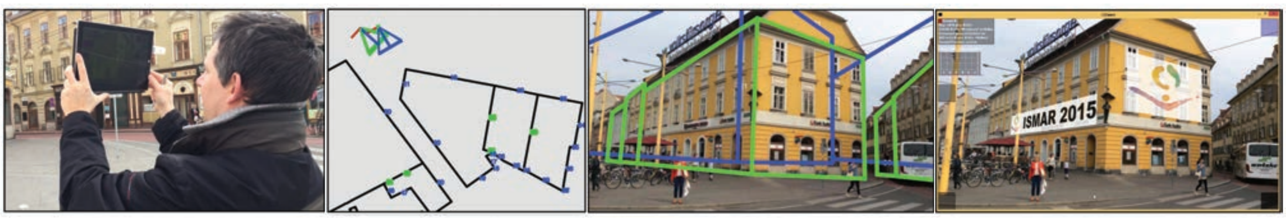
In "Instant Outdoor Localization and SLAM Initialization from 2.5D Maps" (*IEEE Trans. Visualization and Computer Graphics*, vol. 21, no. 11, 2015, pp. 1309–1318), Clemens Arth and his coauthors propose an alternative approach to outdoor localization that works instantaneously, using only the video image, the device sensors (such as GPS and compass), and public OpenStreetMap (www.openstreetmap.org) data. The map data, consisting of building outlines and roof height, is freely available for most cities.

Arth and his coauthors began with a rough location estimate from the device sensors and walked through a series of steps to match the vertical and horizontal lines delineating facades in the video image with a 3D projection of the buildings described in the map data (see Figure 1). With this minimal information, they were able to obtain a localization that was nearly as accurate as that achieved with state-of-the-art image-matching methods.

This approach represents a significant step toward making outdoor localization on mobile devices practical, and expensive, citywide 3D reconstruction optional. The required map data is already available and has lightweight storage requirements. Because fewer database items are involved than when 3D reconstructions are used, computation can be optimized to operate on low-powered mobile devices that potential users already own. ■

LEILA DE FLORIANI is a professor of computer science at the University of Genoa and an adjunct professor at the University of Maryland, College Park. Contact her at ldf4tvcg@umiacs.umd.edu.

See **www.computer.org/computer-multimedia** for multimedia content related to this article.



(a) (b) (c) (d)

Figure 1. Instant outdoor localization that uses only the video image, device sensors, and OpenStreetMap data. (a) User with a mobile device. (b) OpenStreetMap data. (c) Refining the estimated location from raw GPS data (blue) toward the observed building outlines (green). (d) Presenting augmentations registered in 3D.

Magazine Roundup

The IEEE Computer Society's lineup of 13 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 cloud migration and microchip manufacturing. Here are highlights from recent issues.

Computer

Computer's March 2016 special issue explores **communications and privacy under surveillance**, which has become an increasing concern as technology advances have made electronic snooping easier and more effective.

IEEE Software

Software engineering for big-data systems is complex and faces challenges including pervasive distribution, write-heavy workloads, variable request loads, computation-intensive analytics, and the need to provide high availability. *IEEE Software*'s March/April 2016 special issue examines several facets of this complicated puzzle.

IEEE Internet Computing

IEEE Internet Computing's March/April 2016 issue looks at what's on the horizon for **tomorrow's Internet**. Representing a balance of emerging and maturing technologies and applications, the articles showcase a mix of research with broad coverage in information privacy, cloud computing, service-oriented computing, internetworking, and social-network analysis.

Computing in Science & Engineering

CiSE's March/April 2016 special issue includes five of the best papers from the **RESPECT (Research on Equity and Sustained Participation in Engineering, Computing, and Technology) 2015 conference**. The articles are part one of a two-part series addressing research on broadening participation in computing at all levels of education, focusing on gender, race, and ethnic diversity.

IEEE Security & Privacy

Vulnerabilities, whether deliberate backdoor access mechanisms or accidental flaws, make us all less secure. Getting security right is harder than it looks, and the best chance to do this is to make the **cryptography** as simple and public as possible, noted Bruce Schneier in “Cryptography Is Harder than It Looks,” from *S&P*’s January/February 2016 issue.

IEEE Cloud Computing

With cloud computing emerging as one of the enterprise’s key enabling technologies, a fundamental challenge becomes simply **understanding complex cloud architectures**, how they work, and the value they can bring. “Understanding Complex Cloud Patterns,” from *IEEE Cloud Computing*’s January/February 2016 special issue, helps readers choose from among the available architectures.

IEEE Computer Graphics and Applications

Many countries’ economic advances are due to technological developments, which often rely on informatics that encompass data, knowledge, information systems, and information and communication technologies. Visual computing plays an important role in this process, as highlighted by *CG&A*’s March/April 2016 special issue on **visual computing and the progress of developing countries**.

IEEE Intelligent Systems

IEEE Intelligent Systems’ March/April 2016 special issue discusses improvements in **pattern recognition** theory and applications, particularly those involving pattern classification and image analysis. The articles in part one of this series address subjects such as image feature representation and contextual pattern analysis.

IEEE MultiMedia

In a world of big data, sometimes having the right amount of small data at the right time can pack a powerful punch, a concept explored in “**Small Data, Big Impact**,” which appears in *IEEE MultiMedia*’s January–March 2016 issue. The authors present their research project in this area, its multistage design process, and their lessons learned.

IEEE Annals of the History of Computing

Most doctors’ offices have computers, but little communication is taking place among the machines. “Ask Your Doctor … About Computers,” from *IEEE Annals*’ January–March 2016 issue, surveys efforts to introduce **computerized, centralized, and portable electronic health record-keeping systems** in the US and explains why they have been largely unsuccessful.

IEEE Pervasive Computing

The **pervasive computing and Internet of Things communities** examine similar problems and face similar challenges. “Pervasive Computing and the Internet of Things,” which appears in *IEEE Pervasive Computing*’s January–March 2016 issue, encourages the two communities to work together to achieve common goals.

IT Professional

Since its founding, NASA has led the US government’s exploration of space, and its advances have benefited both government and the private sector. Because NASA has unique requirements for data collection and analysis, it has been on the leading edge of many data, computational science, and other innovations, notes “**Really, Really Big Data: NASA at the Forefront of Analytics**,” from *IT Pro*’s January/February 2016 issue.

IEEE Micro

Computing in large-scale systems is shifting from the traditional compute-centric model to a data-centric one. This trend is being driven by today’s large data volumes and the loss of performance and energy efficiency caused by the typical approach of moving data to the locations where computations are performed. **Near-data processing**—the topic of *IEEE Micro*’s January/February 2016 special issue—solves this problem by moving the computation to the data.

Computing Now

The Computing Now website (http://computingnow.computer.org) features **up-to-the-minute computing news** and blogs, along with articles ranging from peer-reviewed research to opinion pieces by industry leaders.

Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.

EDITOR'S NOTE

Advancing the Internet of Things

In the technology revolution that is the Internet of Things (IoT), everyday objects—such as thermostats, cars, consumer-electronic devices, home appliances, factory equipment, vending machines, and building-security products—have embedded Internet connectivity. This allows them to send and receive data, and communicate with one another or with centralized control systems.

The IoT lets systems remotely and automatically both sense and control objects across the Internet, integrating the physical world into computer-based systems and promising greater efficiency, accuracy, and cost savings.

However, the IoT also faces several significant challenges, including security, privacy, scalability, design complexity, and concerns about safety resulting from the lack of human control over systems. This *ComputingEdge* issue examines the IoT's technology, opportunities, and challenges.

Developers must consider the dependencies implicit in their IoT designs, factoring in the resilience needed to keep the infrastructure operational, according to Internet pioneer Vinton Cerf in *IEEE Internet Computing*'s "Dependencies."

A major threat to the IoT's effectiveness is its lack of architecture standards and connectivity. *IEEE Software*'s "Reference Architectures for the Internet of Things" reviews recent IoT architecture evolution and what it means for industry-based projects.

Focusing on a particularly security-sensitive product—the garage-door opener—*IEEE Security & Privacy*'s "Garage Door Openers: An Internet of Things Case Study" investigates how IoT devices might be designed with security in mind.

The author of *IEEE Pervasive Computing*'s "Pervasive Computing and the Internet of Things" encourages the pervasive computing and IoT communities to join forces because they share largely overlapping technical interests and goals.

ComputingEdge articles on other subjects include the following:

- *IT Professional*'s "IT Project Failures: What Management Can Learn" provides suggestions about how managers can use the lessons of projects gone wrong to ensure future success.
- In *IEEE Pervasive Computing*'s "Turning Back the Clock," the author revisits pervasive computing's burning issues from a decade ago and looks at where we stand on them today.
- "Ask Your Doctor … About Computers," from *IEEE Annals of the History of Computing*, briefly surveys efforts to introduce computerized, centralized, and portable electronic health records in the US and why they have been largely unsuccessful.

Dependencies

Vinton G. Cerf • *Google*

A couple of years ago at a Marconi Symposium (www.marconisociety.org) I was lecturing about information infrastructure in San Diego. During this lecture, I mentioned that as dependent as we are on communication technology, we're even more dependent on electricity. A few hours later, all of the power went out in San Diego and beyond for a couple of days. After the power finally came back on, I got an email from one of the attendees saying, "You didn't have to do that to make your point!"

The fact is, we're incredibly dependent on electricity and the things that use it. Last month, I wrote in a *Communications of the ACM* column ("Cerf's Up") about my personal dependency on hearing aid batteries (see http://cacm.acm.org/magazines/2015/5/186028-cascade-failure/fulltext). These expensive devices are worthless if I can't get batteries for them. Readers are surely familiar with the proverb "For Want of a Nail."

Technological progress seems to weave its own Web of dependencies and sometimes we don't think enough about the implications. For electricity, battery backup and generators are becoming more common. Elon Musk recently announced a product line of home battery backups that complement the batteries that run his now-famous Tesla automobile, for example. In fact, we can imagine that power generation will become more distributed, making us less dependent on central power generation and wired distribution and more dependent on solar or other renewable power-generation options. This is a complex area and it isn't easy to get it right.

In Hawaii, for example, it's increasingly common for homes to install solar power arrays and sell power back to the electric companies. The companies must make investments in new power management equipment and facilities, however, and this costs money that they can't afford if they're buying power from residents who are paying them less and less. Getting the incentives and business models right is part of the optimization picture.

Thus, incentives form an interesting part of an equation that balances dependencies and backup. I think the same concerns may arise as the Internet of Things (IoT) becomes a more common reality. We have many aspirations arising from the introduction of software and communication into ordinary appliances. Already we see one form of this dependency in the programmable chips that run automobile engines and other functions now common in automobiles. If the software is inaccessible to all but certified maintenance staff, the long-term prospects for maintaining these vehicles may diminish. We can appreciate that access to this software needs to be constrained—you don't want hackers to modify your automobile controls to create hazardous conditions or deliberately cause damage. At the same time, we can imagine automobiles and appliances rendered useless, despite their cost, if the software can't be accessed for repair. "Sorry, we no longer support the 2023 version of the operating system on your Phlogiston Phantasy. Too bad if it's stuck in the garage."

As the IoT ecosystem evolves, it seems important to ask questions about the dependencies implicit in the design and resilience of the infrastructure needed to keep them operating. We can also ask what happens when they're prevented from operating because of *their* dependency on electrical power, the Internet, mobile phones, and so on.

The same questions have no doubt been asked about "fly-by-wire" designs, and usually the answer is redundancy. Another kind of answer might be rapid recovery, and redeployment of equipment and facilities. A third answer might

 2469-7087/16/$33.00

be a backup mechanical system — although for many use cases, this probably won't be an alternative. The whole point of IoT, in many instances, is precisely the software's flexibility.

Speaking of which, this draws our attention to one more dependency: software has bugs and we need a way to update software safely. That is, to be assured the updates can happen and come from a trusted source. Many devices that are part of the IoT zoo might not have displays or other mechanical interfaces. Our mobiles and Bluetooth radios might be the only way to reach inside for control. Strong authentication methods will be attractive, to limit access and to assure that the updates come from recognized sources. Of course, any time we rely on strong authentication mechanisms we also inherit potential, if inadvertent, denial-of-service situations if the authentication fails. "Sorry, your refrigerator update can't be installed; you haven't provided the proper access key. Too bad if your ice cream is melting."

I don't have a lot of answers for you here, but I hope readers who are involved in IoT will give serious thought to dependencies in the infrastructure needed to make them work.

Vinton G. Cerf is vice president and chief Internet evangelist at Google, and past president of ACM. He's widely known as one of the "fathers of the Internet." He's a fellow of IEEE and ACM. Contact him at vgcerf@gmail.com.

This article originally appeared in IEEE Internet Computing, *vol. 19, no. 4, 2015.*

Editor: **Christof Ebert**
Vector Consulting Services
christof.ebert@vector.com

Reference Architectures for the Internet of Things

Michael Weyrich and Christof Ebert

The Internet of Things (IoT) is about innovative functionality and better productivity by seamlessly connecting devices. But a major threat is the lack of architecture standards for the industrial Internet and connectivity in the IoT. Here, Michael Weyrich and I review recent IoT architecture evolution and what it means for industry projects. I look forward to hearing from both readers and prospective column authors. —*Christof Ebert*

IMAGINE BEING the engineering head of an automotive-engine manufacturer. Having been making engines over the past decades, your company knows them inside and out. But engines are changing quickly—too fast for many in your company, be it in manufacturing, repair, or service. First, embedded electronics made your engines anything but mechanical. Then, the engines became increasingly connected to other electronic systems in the car. Suddenly there is an explosion of technological opportunities and challenges. You need a device cloud for different configurations and their updates, integration and test capabilities to handle the state explosion of your systems, security skills to keep the machines in a defined mode and thus safe for the public, new tablet applications for diagnosis and repair, connectivity to enterprise-resource-planning systems, and big-data analytics to support engineering and maintenance throughout the life cycle.

The upside is obvious. Knowing typical user needs and orchestrating them in the products and workflows in real time creates a wealth of value:

- *Flexibility*. Products can autonomously adapt to usage scenarios such as assisted living, intelligent buildings, smart transportation, energy, healthcare, transportation, or entire supply chains.
- *Usability*. Even complex products can be operated more easily, thus improving the user experience while mitigating hazards.
- *Productivity*. Service extends toward predictive maintenance and proactive enhancements, improving uptime and thus productivity.

The Internet of Things (IoT) will boost a tremendous amount of innovation, efficiency, and quality. Connecting production, medical, automotive, or transportation systems with IT systems

 Published by the IEEE Computer Society

and business-critical information will provide tremendous value to organizations. Major IT companies such as Cisco and SAP have predicted billions of networked devices and a universe of IT-based business services, with expectations of a trillion-dollar business.[1,2]

But who in your company understands the combined software and IT needs and necessary architectures and technology stack? Not many. Business leaders know the value chain but don't bother about technology. Manufacturing shies away when confronted with software technology. IT departments tend to overlook in their big and distant perspective that there are real products and customers. Engineering departments focus on system development and embedded electronics, and consider IT one of those big things that never work as expected. IoT architecture and modeling solutions must connect heterogeneous communities to understand and work together.

Software engineering for the IoT poses challenges in light of new applications, devices, and services. Mobility, user-centric development, smart devices, e-services, ambient environments, e-health, and wearable or even implantable devices pose specific challenges for specifying software requirements and developing reliable, safe software. Specific software interfaces, agile organization, and software dependability require particular approaches for security, maintainability, and sustainability. Reference architectures for the IoT aim to help developers meet these challenges.

A Brief History of the IoT

The term "Internet of Things" was first used in 1999 by Kevin Ashton, who worked on a standard for tagging objects using RFID for logistics applications. However, the idea of ubiquitous computing goes back to the late 1980s. Since then, researchers have worked on many systems focusing on tags and sensor networks, middleware and cloud technologies, and communication networks.

A highly visible milestone was reached when the Internet Engineering Task Force released IPv6, the protocol that enables the IoT. Recently, the IoT has received a boost from commercial engagement and work on reference architectures driven by the major industries:

- Google has announced Brillo as an OS for IoT devices in smart homes.
- Devices are commercially available for machine-to-machine (M2M) communication standards such as Bluetooth, ZigBee, IPC Global's standards, and low-power Wi-Fi.
- Microsoft has announced that Windows 10 will support embedded systems for widespread microcontrollers such as Raspberry Pi 2.
- Samsung and other companies have announced a new generation of chips for smart devices.
- Many implementation reports have described networked microcontrollers serving as hubs for sensors, actuators, and tagging.

> The IoT has become the major disruptive technology changing software—and society.

Two alternative names are used for the IoT: Industrial Internet in North America and Industry 4.0 in Europe. However, the preferred name seems to be the Internet of Things.

IoT Architecture and Model Evolution

So far, IoT applications have been based on fragmented software implementations for specific systems and use cases. The big need for reference architectures in industry has become tangible with the fast-growing number of initiatives working toward standardized architectures. These initiatives aim to facilitate interoperability, simplify development, and ease implementation. Table 1 gives a brief overview of them.

For example, the Reference Architecture Model Industrie 4.0 (RAMI 4.0) goes beyond the IoT, adding manufacturing and logistics details. The Industrial Internet Reference Architecture (IIRA) has a strong industry focus. The Internet of Things—Architecture (IoT-A) provides a detailed view of the IoT's information technology aspects. Major standardization is happening in M2M communication, employing efficient, scalable, and secure communication stacks. This standardization is based on a modified Open Systems Interconnection (OSI) stack and proposes specifications for the

TABLE 1

Internet of Things (IoT) reference architectures are evolving in close collaboration between research and industry.

Category	Initiative	Description	Status	URL
IoT reference architecture models	Reference Architecture Model Industrie 4.0 (RAMI 4.0)	A reference architecture for smart factories dedicated to IoT standards, which started in Germany and today is driven by all major companies and foundations in the relevant industry sectors.	Version 1 as of July 2015	www.zvei.org/en/association/specialist-divisions/automation/Pages/default.aspx
	Industrial Internet Reference Architecture (IIRA)	The Industrial Internet Consortium (founded by AT&T, Cisco, General Electric, IBM, and Intel) has delivered a reference architecture for broader consideration and discussion.	Version 1.7 as of June 2015	www.iiconsortium.org
	Internet of Things—Architecture (IoT-A)	The IoT-A delivered a detailed architecture and model from the functional and information perspectives. The project also performed a detailed analysis of system requirements.	The final architectural reference model for the IoT v.3.0 as of July 2013	www.iot-a.eu/public/public-documents/d1.5/view
	Standard for an Architectural Framework for the Internet of Things (IoT)	The IEEE P2413 project has a working group on the IoT's architectural framework, highlighting protection, security, privacy, and safety issues.	An ongoing activity, with no white papers or defining documents as of Sept. 2015	https://standards.ieee.org/develop/project/2413.html
	Arrowhead Framework	This initiative enables collaborative automation by open-networked embedded devices. It's a major EU project to deliver best practices for cooperative automation.	Ongoing updates and release of material by spring 2016	www.arrowhead.eu
Machine-to-machine (M2M) standards relevant to the IoT	European Telecommunications Standards Institute Technical Committee (ETSI TC) for M2M	The TC provides IoT communication standards.	Various available standards and drafts	www.etsi.org/technologies-clusters/technologies/m2m
	International Telecommunication Union Telecommunication Standardization Sector (ITU-T)	The ITU-T has coordination activities on aspects of identification systems for M2M.	Various available standards and drafts	www.itu.int/en/Pages/default.aspx
Further activities	European Research Cluster on the Internet of Things (IREC)	The IREC is involved in many IoT-related issues, including connected objects, the Web of things, and the future of the Internet.	Ongoing updates	www.internet-of-things-research.eu
	Smart Appliances (SMART) study	This EU-funded study focused on semantic assets for smart-appliance interoperability.	Smart-appliance reference ontology definition as of Mar. 2015	https://sites.google.com/site/smartappliancesproject/home or http://ontology.tno.nl/saref

data link, adaptation, network, and transport layers.

What Are Reference Architectures About?

Identifying and structuring an architecture or model is a long, tedious process with much negotiation to abstract from specific needs and technologies. Such a reference can serve as an overall, generic guideline; not all domain applications will require each detail for real-life implementation.

Nevertheless, the requirements are easy to grasp:

- Connectivity and communications are important. This might involve one-to-one connectivity (unicast) or data collection and information dissemination to multiple partners (multicast and anycast).
- Device management must provide solutions once a device is added or a device configuration changes and must be propagated to other devices.
- Data collection, analysis, and actuation are relevant for extracting information and knowledge for offering services.
- Scalability is important to handle increased processing volumes for different installation sizes.
- Security features are necessary to provide trust and privacy and are required for all aspects of the IoT.

An IoT reference architecture handles those requirements and forms a superset of functionalities, information structures, and mechanisms. Additional consideration of entities and their interaction leads to a reference model. Such a model integrates aspects of the related entities such as human users, device implementations, and server structures and provides a more complete view of, or templates for, the overall setup and its domain implementation. Both the architecture and model help describe and map technologies to business cases.

Available Architectures

Two major architectures are available: the IoT-A and IIRA. Both architecture proposals have been thoroughly prepared, but the IoT-A has been described in detail and extended. Since its 2012 launch, it has been synchronized with the IoT community and incorporates multiple views. In contrast, the IIRA still aims for feedback and further detailing.

We now compare these architectures regarding their capabilities and layers according to three perspectives.[3]

The first perspective is semantic orientation—the interpretation of data and information to create knowledge for business cases. The IoT-A concentrates on the generic aspects of informatics instead of the application facets of semantics. In contrast, the IIRA focuses on the functionality of the industry domain, such as business, operations (prognostics, monitoring, optimization, and so on), information (analytics and data), and application (UIs, APIs, logic, and rules). RAMI 4.0, which is domain specific, extends the view of the IIRA toward the life cycle and value streams of manufacturing applications.[4] In particular, it enhances the functional-layer structure by two dimensions—the life cycle and value stream—as well as by hierarchy levels (for further details, see International Electrotechnical Commission [IEC] standards 62890, 62264, and 61512).

The second perspective is Internet orientation and has two aspects. The first is middleware for service support and data management in the cloud and servers. The IoT-A extensively covers the modeling and structuring of IoT business process management, virtual entities, IoT services, and cross-service organization from the functional, information, and domain viewpoints, in an abstract manner. Cloud aspects, in the sense of the server-side architecture and its management, are defined by the implementation. The same applies to agents and code on domain-specific devices. The IIRA also focuses on these aspects but remains closer to business and use cases.

The second aspect is networking, transport, and data links. Both architectures briefly regard these things but refer mostly to M2M communication to cover the OSI stack's lower layers. For instance, the network layer could be implemented by IPv6, whereas network and transport could be based on UDP (User Datagram Prototcol) and CoAP (Constrained Application Protocol). Alternative realizations might employ the lightweight protocol MQTT, which could be used on top of TCP/IP instead of HTTP.

The third perspective is things orientation, which centers on assets such as sensors, actuators, and tags, which are crucial in both the IoT-A and IIRA. This is the classic approach for the automation industry, which tries to define bottom-up a reference around tangible objects and their individual data sources and information needs.

Both architectures have management and security mechanisms across all layers. The architecture proposals help define and explain

TABLE 2 IoT architectural layers and protocols structured in three perspectives.

Perspective	Application
Semantic oriented	Service protocols such as OPC UA (OPC Unified Architecture), UPnP (Universal Plug and Play), DPWS (Devices Profile for Web Services), CoAP (Constrained Application Protocol), and EXI (Efficient XML Interchange).
Internet oriented	Interconnectivity and protocol conversion based on UDP (User Datagram Protocol) vs. TCP with HTTP or MQTT. Support for IPv4 or IPv6.
Things oriented	A physical layer and data link layer with low-level communication protocols suitable for easy installation and maintenance.

the IoT's overall structure. They provide descriptive models of how IoT devices and humans interact and process data, incorporating patterns of M2M communication standards. These models have different perspectives and granularities in describing IoT. Table 2 shows the most recent architectural layers and protocols.

Deployment Hurdles

The IoT is rich in approaches, concepts, and structures. Various initiatives have already delivered IoT models, architectures, and tools. However, a further convergence of approaches and industrial standards is required for simplification. This raises issues about the key IoT enablers.

One key enabler is efficient, secure communication. Many technologies exist to implement the communication stack. Low-power wireless networks that require little implementation effort are necessary, for which security is an important factor.

Device hardware suitable for implementing the IoT is readily available. However, it's questionable whether device-oriented, real-time networked OSs will ever exist for the different application domains or whether the diversity of real-time OSs will continue.

Logic controllers, cyclic or event-based, have been extensively discussed and standardized. Typical use cases have been identified, and solutions are available, which could be reconsidered and adopted for the IoT. Conversely, the industrial-automation community might profit from new development kits for the IoT.

Finally, big-data analytics and the human–machine interface form the front end for users and are close to the business case. Owing to applications' individual characteristics, not much standardization is available yet. However, a first step toward standardization could be semantic description. Standardized languages such as Vorto or Weave are important to describe devices, parameters, user interfaces, and so on.

The IoT has become the major disruptive technology changing software—and society. Businesses, cities, and even countries must now start on this networked digitization. But focusing only on networking isn't enough. Networking depends on the software that collects the information, and that software must be designed for the IoT's unique needs. Now is the time for software and industry leaders to agree on standards and thus avoid systems that are crippled because they can't talk with each other.

References

1. R. Howells, "The Business Case for IoT," SAP, 18 June 2015; http://scn.sap.com/community/business-trends/blog/2015/06/18/the-business-case-for-iot.
2. D. Evans, "The Internet of Things: How the Next Evolution of the Internet Is Changing Everything," Cisco, Apr. 2011; www.cisco.com/web/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf.
3. L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A Survey," *Computer Networks*, vol. 54, no. 15, 2010, pp. 2787–2805; www.sciencedirect.com/science/article/pii/S1389128610001568.
4. *Status Report: Reference Architecture Model Industrie 4.0 (RAMI4.0)*, VDI/VDE Society Measurement and Automatic Control, July 2015; www.vdi.de/fileadmin/vdi_de/redakteur_dateien/gma_dateien/5305_Publikation_GMA_Status_Report_ZVEI_Reference_Architecture_Model.pdf.

MICHAEL WEYRICH is the director of the University of Stuttgart's Institute for Automation and Software Systems. Contact him at michael.weyrich@ias.uni-stuttgart.de.

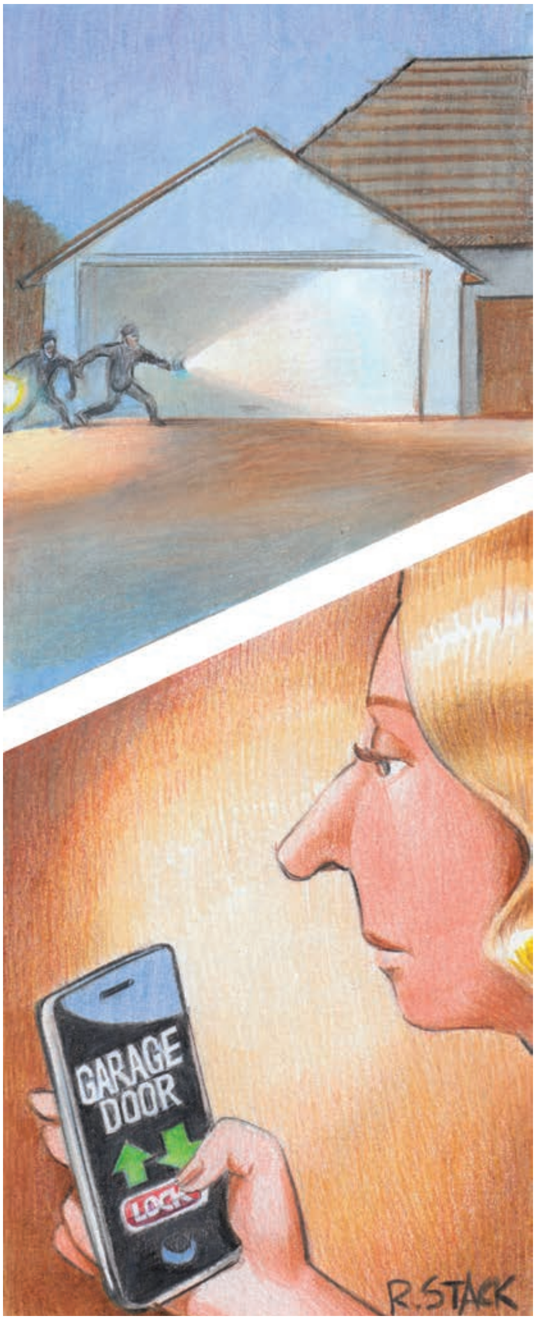
CHRISTOF EBERT is the managing director of Vector Consulting Services. He is on the *IEEE Software* editorial board and teaches at the Universities of Stuttgart and Paris. Contact him at christof.ebert@vector.com.

This article originally appeared in IEEE Software*, vol. 33, no. 1, 2016.*

Editor: Jonathan Margulies, jonathan@qmulos.com

Garage Door Openers: An Internet of Things Case Study

Jonathan Margulies | Qmulos



Early last year, my garage door opener's motor died. While researching potential replacement units, I focused on Chamberlain's products because they had a reputation for high quality. Once I settled on a model, I noticed another option: for a little more money, Chamberlain would include the MyQ Internet Gateway, its new system for monitoring and controlling the opener via the Internet. Curiosity got the best of me, so I went for it.

After installing the opener, the MyQ languished in my closet for months. I loved the idea of getting an alert if I left my garage door open—I can't count the number of times I've turned around five minutes after leaving the house to double-check that it was closed—but I felt sure there would be a security flaw in the MyQ that would make me worse off. The emergence of the Internet of Things (IoT) has turned trusted, long-standing companies into unwitting network attack vectors.[1]

But the MyQ is different from—and more impactful than—other IoT devices: it controls access to my house. That got me thinking: What if I wanted to solve this problem from the ground up? How would I design an Internet-connected garage door opener ("IoT opener") to be adequately secure? Is it possible?

The Ground Rules

First, I define "adequately secure" to mean no less secure than the traditional rolling-code garage door opener. Because that's the system the MyQ is replacing, it seems like the right standard.

Second, due to space constraints, I focus only on design. The usual implementation caveats—for example, the need for well-written code and correct use of encryption libraries—still apply, but I won't address them in detail.

Third, I assume the same basic set of features the MyQ offers: a user can open or close a garage door via the Internet from a smartphone or computer and receive emails or push notifications when the door's status changes.

Standard Garage Door Openers

If the standard is the security of rolling-code openers, we first need to understand how those openers work.

The most popular rolling-code implementation is a product called KeeLoq, a lightweight block cipher that generates codes based on a cryptographic key and a counter (www.webcitation.org/6ZZYZpH2n). When a user syncs a remote control with a garage door opener, the remote control begins to generate the same codes, in the same order, as that opener. Thereafter, when a user pushes the "open" button, the remote control increments its counter, generates a new code, and broadcasts that code wirelessly. When the opener receives a code, it checks the code against the next 256 codes in its

 Published by the IEEE Computer Society

queue. (Checking against so many possible codes helps ensure that the remote control and the opener don't lose sync when a user presses the button outside the opener's receiving range.) If the code is a match, the opener increments its counter to just above the matching code and opens the door. In addition to using remote controls, some users mount keypads in front of their garages that similarly sync with the openers; these keypads broadcast a code when a user correctly enters a numeric password.

The simplest way for attackers to open a rolling-code garage door opener is to sync it to a new remote control. Replacement remote controls are available at just about any hardware store, and syncing them requires only a few minutes alone in the garage. A similarly easy option is to go after the keypad by spying on the user, or deduce or brute-force the code. A third option is a physical attack. Most openers include an emergency release rope just inside the door. If an attacker can slip a wire hanger above the door and latch onto that rope, a skilled tug can unlock the door. The final option for attacking traditional openers is to go after the rolling-code mechanism itself. Over the past decade, several researchers have developed methods to derive a KeeLoq key given access to a working, synced remote control.[2–4] A simpler but less effective approach is to sniff a code over the air from a remote control by pushing the "open" button outside the opener's range, and then using that code before the owner comes home (at which point, that code will expire).

All these attacks require close proximity to either the garage or the remote control and are sufficiently difficult that virtually all intruders prefer to break a window, force a door open, or pick a lock. But where are those intruders? Why aren't they taking advantage of the universally weak security of modern suburban homes? As podcaster Roman Mars eloquently observed, "locks have become a social construct as much as they are a mechanical construct" (http://99percentinvisible.org/episode/perfect-security). Garage door openers only need to be secure enough to let passersby know we don't want them to come in.

Openers and the Internet of Things

Exposing garage door openers to the Internet might make them such easy targets as to pose a real risk. What if an attacker could indiscriminately send open commands to any opener?

Exposing garage door openers to the Internet might make them such easy targets as to pose a real risk.

What if every time an email account is hacked, the hacker is given a clear path to find the user's home address and credentials to open that user's garage? What if an attacker managed to download a whole database of user credentials for IoT openers? Any of these possibilities would make home intrusions so easy as to be inevitable. We're starting to see this evolution with cars that use Bluetooth keys; it's become so easy and cheap to break into some of them that insurers have started demanding additional security measures.[5]

But before I can delve deeper into how those types of attacks might happen on IoT openers, I must first address a key architectural question: Will the opener authenticate and authorize the user, or will a cloud service do so on the opener's behalf? When I bought the MyQ, I hoped it would be the former, as I had grand plans for putting the opener on its own private virtual LAN and having my wife use a virtual private network to reach it, because who would trust a garage door opener company with network access control? But I was naive.

The IoT industry has clearly decided that having a central service act as a clearinghouse for authentication, authorization, and commands is a must, and it's easy to see why: it frees them from having to worry about configuring home routers, setting up dynamic DNS for when customers' IP addresses change, or having access to all of a user's relevant data when those inevitable tech support calls come in. The problem is that the cloud service opens another attack surface, and a big one: instead of having to hack a single IoT opener at a time, attackers can try to hack them all through the cloud service. It's a single point of failure for authentication, integrity, and availability. Indeed, MyQ experienced an unplanned four-hour disruption in late April that affected all users, and I doubt it will be the last.

Cloud Service Authentication

Using a cloud service as a central hub isn't the security decision I would have made, but it seems like such a foregone conclusion that I'll treat it as an assumption for the remainder of this article. Authentication at both the cloud service and the opener is of paramount concern. The obvious way for the cloud service to authenticate to the opener is with a certificate, but what happens when attackers compromise the private key behind that certificate? This could allow attackers to send arbitrary commands to any opener, or something much worse: if the certificate can be used to force over-the-air software updates, attackers

Chamberlain Statement

The following statement was sent to the author on 12 June 2015:

> Chamberlain built its MyQ technology—as we have all of our products for more than 40 years—on a foundation of safety and security. We have an aggressive product roadmap that includes continuous security updates, including the feature recommendations noted in the IEEE [*Security & Privacy*] story.
>
> Specifically, Chamberlain will upgrade password requirements in the second half of this year [2015], and is looking at the best ways to implement two-factor authentication based on our user needs and their usage scenarios. We also plan to introduce multi-user access so that account administration details and log-in credentials are secure to only one account owner, while allowing garage door access to other users. This will include assigning different levels of permissions based on user access level, times of day, days of week, etc.
>
> In addition to the specific updates noted above, we combine our own team's expertise in security technologies with reputable third-party security firms to audit our systems on an ongoing basis. Our continuous security updates and processes include using industry standard encryption, applying the latest security techniques, and periodic security testing with respected outside services. We also recommend IoT technology leaders continuously advise their customers on how to maximize the security of their home Wi-Fi network, which are critical gateways to device security for consumers. Chamberlain's brand and reputation are built on a heritage of delivering safe and secure products to consumers; we take the safety and security of the smart home very seriously.

could gain control of the whole system. For this reason, software updates should be user initiated and openers should regularly check for certificate revocation.

Authentication at the cloud service is more complicated. Perhaps the worst-case scenario is when attackers download the password database, as in the famous attacks on Sony's PlayStation Network (www.cnet.com/news/playstation-network-still-offline-after-suspected-attack) and LinkedIn (http://money.cnn.com/2012/06/06/technology/linkedin-password-hack). If the passwords aren't sufficiently complex to stymie brute-force attacks or aren't encoded by an adequate key derivation function (such as bcrypt), user account takeover becomes trivial.

What about password reset, the issue behind attacks on countless celebrities' email and Apple iCloud accounts? Google recently published research suggesting that security questions are insufficient to protect accounts.[6] As of this writing, MyQ uses email for password reset, which, in this case, seems like a terrible idea: any time attackers hijack an email account, they can search for emails containing the term "MyQ" to determine whether the user has a MyQ account, and then search for shipping information to determine the user's home address. Ironically, just about the only personal information the MyQ website asked for was my home address, which is the information they should least want to have on file. This combination of information allows attackers to build databases of locations of vulnerable openers.

An opener is actually an interesting case from a password reset perspective in that it has an unusual security feature: it never moves. That means a user can't lose it, and it would therefore be reasonable for Chamberlain to require users to electronically prove they have possession of it (for example, the opener could display a code that rolls every few seconds). Such a feature would also help new homebuyers prove transfer of opener ownership.

Potential Security Improvements

The other side of this discussion is the security improvements networked openers might offer. One improvement is *two-factor authentication* (2FA). Many of the problems outlined in this article can be mitigated by 2FA and, in a system that's already so reliant on smartphones, users are already carrying the obvious second factor in their pockets. A second improvement is policy-based access control. This can be useful in several ways:

- allowing multiple user accounts to control the door, but only one to administer it;
- allowing administration from specific devices only;
- restricting certain accounts (such as caregiver or contractor) to operating the door only during business hours; and
- creating time-limited guest accounts.

These policies could be easy to administer through a Web interface.

This article originally appeared in
IEEE Security & Privacy, *vol. 13, no. 4, 2015.*

A third improvement is a more granular alerting system. I'd be much more interested in knowing that a new user or remote control was given access to my opener, or that the door opened in the middle of the night, than in knowing that my wife opened the door at 5 pm.

Convenience will continue to drive companies that lack information security expertise to build IoT devices, and consumers to buy them. Ideally, this new IoT world would be built on a few competing platforms by people who understand and can address the security risks—Apple, Google, and Facebook have all recently launched the beginnings of IoT platforms—rather than inexperienced companies rolling their own authentication, authorization, and communication code. Although the risks and functions change from one IoT device to another, they all need the same basic security infrastructure: a way for users to authenticate, two-way mapping between users and devices, access policy creation and enforcement, logging and alerting capability, and secure communication. These problems are largely solved; they just need to be made easy for IoT developers to use. When that happens, and the right security features are in place, I think IoT openers could become difficult enough to attack that no one would bother with them, just like the openers most of us have today. In the meantime, my MyQ is going back in the closet. ■

Author's Note

I contacted Chamberlain after writing this article, and a representative responded with an outline of plans for addressing some of the concerns in this article. See the sidebar for the response.

References

1. J. Fontana, "Belkin Patches Vulnerabilities in WeMo Devices," ZDNet, 19 Feb. 2014; www.zdnet.com/article/belkin-patches-vulnerabilities-in-wemo-devices.
2. S. Indesteege et al., "A Practical Attack on KeeLoq," *Proc. 27th Ann. Int'l Conf. Theory and Applications of Cryptographic Techniques* (EUROCRYPT 08), 2008, pp. 1–18.
3. A. Bogdanov, "Cryptanalysis of the KeeLoq Block Cipher," Cryptology ePrint Archive, report 2007/55; https://eprint.iacr.org/2007/055.pdf.
4. I. Sheetrit and A. Wool, "Cryptanalysis of KeeLoq Code-Hopping Using a Single FPGA," Cryptology ePrint Archive, report 2011/242; https://eprint.iacr.org/2011/242.pdf.
5. H. Osborne, "Thieves Target Luxury Range Rovers with Keyless Locking Systems," *Guardian*, 27 Oct. 2014; www.theguardian.com/money/2014/oct/27/thieves-range-rover-keyless-locking.
6. J. Bonneau et al., "Secrets, Lies, and Account Recovery: Lessons from the Use of Personal Knowledge Questions at Google," Google Research Archive, report 43783; http://research.google.com/pubs/pub43783.html.

Jonathan Margulies is the chief technology officer at Qmulos. Contact him at jonathan@qmulos.com or follow him on Twitter @unsaltedhash.

Editor in Chief: Maria R. Ebling ■ IBM T.J. Watson Research Center ■ ebling@us.ibm.com

Pervasive Computing and the Internet of Things

Maria R. Ebling, *IBM T.J. Watson Research Center*

This issue marks a change that has been in the works for 18 months. In case you didn't notice, our subtitle has changed from "Mobile and Ubiquitous Computing" to "Mobile Systems | Ubiquitous Computing | Internet of Things." How does IoT differ from pervasive computing, and why did we add it to our subtitle?

INTERSECTING COMMUNITIES

According to Webopedia.com, pervasive computing, also called ubiquitous computing, "is the idea that almost any device, from clothing to tools to appliances to cars to homes to the human body to your coffee mug, can be imbedded [sic] with chips to connect the device to an infinite network of other devices…. in such a way that the connectivity is unobtrusive and always available."

The origins of *IEEE Pervasive Computing* are in realizing the vision outlined in Mark Weiser's seminal *Scientific American* paper, "The Computer for the 21st Century," published in 1991. This vision is summarized by his famous opening statement: "The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it." Mahadev Satyanarayanan outlined this origin in our inaugural issue,[1] and we revisited our progress toward Weiser's vision in a special issue[2] recognizing the 20th anniversary of Weiser's seminal article.

Webopedia.com defines IoT as "the ever-growing network of physical objects that feature an IP address for Internet connectivity, and the communication that occurs between these objects and other Internet-enabled devices and systems." It goes on to say that IoT "extends Internet connectivity beyond traditional devices ... to a diverse range of devices and everyday things that utilize embedded technology to communicate and interact with the external environment, all via the Internet."

Both communities are pursuing similar use cases, including smart cities, environmental monitoring, agriculture, home automation, and health and wellness monitoring.

Like pervasive computing, the origins of IoT date back to a visionary researcher from the 90s. In 1999, just eight years after Weiser's seminal article, Kevin Ashton, the head of the Auto-ID Center at MIT, made a presentation to Proctor & Gamble entitled, "Internet of Things."[3] In fact, numerous overviews of IoT[4,5] explicitly reference both Weiser's article as well as Ashton's presentation (also see http://postscapes.com/internet-of-things-history). According to Friedemann Mattern and Christian Floerkemeier,

> *The Internet of Things represents a vision in which the Internet extends into the real world embracing everyday objects. Physical items are no longer disconnected from the virtual world, but can be controlled remotely and can act as physical access points to Internet services. An Internet of Things makes computing truly ubiquitous—a concept initially put forward by Mark Weiser in the early 1990s.*[6]

Pervasive computing and IoT examine similar problems and face similar challenges. Some might argue that pervasive computing focuses more on HCI issues—on making the connected things disappear from human attention—while IoT focuses more on connecting the devices. Yet the IoT community also focuses on HCI issues, and the pervasive community thinks about connecting devices too. Both communities are interested in issues beyond just technology, such as privacy, security, and ethics. Both communities are pur-

MISSION STATEMENT: *IEEE Pervasive Computing* is a catalyst for advancing research and practice in mobile and ubiquitous computing. It is the premier publishing forum for peer-reviewed articles, industry news, surveys, and tutorials for a broad, multidisciplinary community.

 2469-7087/16/$33.00

suing similar use cases, including smart cities, environmental monitoring, agriculture, home automation, and health and wellness monitoring.

The communities behind pervasive computing and IoT share the same (or at least largely overlapping) technical interests and goals, so I encourage our readers to participate in IoT forums, and I welcome submissions from the IoT community. I hope that our subtitle change will help our communities join forces and collaborate. We will all make faster progress if we work together toward our common goals.

IN THIS ISSUE

One topic of interest to both communities is that of smart vehicles, so it is highly appropriate that smart vehicle spaces is the theme of our first special issue under our new subtitle. Steve Hodges, Brian Noble, and Venkatesh Prasad have done a tremendous amount of work on this issue, arranging for an interview and a spotlight column in addition to the themed articles. I want to thank them all for their efforts in bringing you this special issue.

In addition to our theme articles, we also have four feature articles and several departments, many of which should be of interest to both the pervasive and IoT communities. In "Soft Actuation: Smart Home and Office with Human-in-the-Loop," Jaroslaw Domaszewicz, Spyros Lalis, Alexsander Pruszkowski, Manos Koutsoubelias, Tomasz Tajmajer, Nasos Grgoropoulos, Michele Nati, and Alexander Gluhak present a system focused on "soft actuation" in which the smart home or office infers a context switch that requires an actuation and triggers its human-in-the-loop to make the actuation. The authors focus on ensuring that the soft actuation remains "calm" by paying careful attention to when to interrupt the user and by providing nonintrusive hints that are easily ignored and sometimes undetected.

Next, Sarah Clinch, Nigel Davies, Mateusz Mikusz, Paul Metzger, Marc Langheinrich, Albrecht Schmidt, and Geoff Ward present "Collecting Shared Experiences through Lifelogging: Lessons Learned." This article describes their Grasmere House experience, in which they collected lifelogging data in an immersive environment that included instrumenting the house as well as deploying wearable, connected cameras over a period of almost three days. One really interesting aspect of this experiment is that they had almost two dozen lifeloggers living in one house! They outline the experiment and explain the many lessons learned. I encourage anyone interesting in lifelogging to have a look at this study.

In "Crowdsensing in the Wild with Aliens and Micropayments," Manoop Talasila, Reza Curtmola, and Cristian Borcea describe two crowdsensing experiments that they recently completed at the New Jersey Institute of Technology. The two experiments involved a mobile, first-person-shooter game—in which users complete sensing tasks by defeating aliens around

UPCOMING ISSUES

Our upcoming issues will focus on the following topics:

- April–June 2016: Domestic Pervasive Computing
- July–September 2016: Pervasive Displays
- October–December 2016: Energy Harvesting and Power Management (www.computer.org/web/computingnow/pccfp4)
- January–March 2017: Drones (www.computer.org/web/computingnow/pccfp1)
- April–June 2017: Smart Cities

EDITORIAL BOARD CHANGES

I would like to thank the following editorial board members for their many years of service and dedication: Elizabeth Belding, John Canny, Hans Gellersen, Sumi Helal, Anthony Joseph, and Rahul Sukthankar. I wish them all the best in their future endeavors.

At the same time, I welcome two new editorial board members: Jesus Favela and Sanjay Sarma. I look forward to their contributions to the magazine.

Jesus Favela is a professor of computer science at CICESE, Mexico, where he leads the Mobile and Ubiquitous Healthcare Laboratory. His research interests include ubiquitous computing, human-computer interaction and medical informatics. Much of his research efforts have focused on the design and evaluation of ambient computing environments for healthcare, mostly to assist the demanding conditions of hospital work and to support older adults and their caregivers. Favela received his PhD from the Massachusetts Institute of Technology (MIT). He is a former President of the Mexican Computer Science Society and the current Dean of Graduate Studies at CICESE. Contact him at favela@cicese.mx.

Sanjay Sarma is the Fred Fort Flowers and Daniel Fort Flowers professor of mechanical engineering and newly appointed Director of Digital Learning at MIT. Sarma was one of the founders of the Auto-ID Center at MIT, which, along with a number of partner companies and its "spin-off," EPCglobal, developed the technical concepts and standards of modern RFID. He also chaired the Auto-ID Research Council consisting of six labs worldwide, which he helped to establish. He is a consultant and board member at several companies, including EPCglobal, and also serves as a permanent guest of the board of GS1 and a member of the board of governors of GS1US. Sarma also serves on the City of Boston's Complete Streets Advisory Group. Sarma received his PhD from the University of California at Berkeley. Contact him at sesarma@mit.edu.

FROM THE EDITOR IN CHIEF

the campus—and a micropayment system—in which participants are paid to complete specific sensing tasks around the campus. The authors describe the two experiments and offer lessons learned about these different approaches to crowdsensing incentives. This article addresses a situation in which the world is not fully instrumented with connected sensors and in which you need to incentivize mobile agents (in this case humans) to augment the sensor network.

Finally, in "Software Support for Multitouch Interaction: The End-User Programming Perspective," Andrea Bellucci, Marco Romano, Ignacio Aedo, and Paloma Díaz present a survey examining the state-of-the-art of software development tools that support the creation of multitouch sensing interactions by end-user programmers. They examine tools ranging from software developer's kits provided to support various mobile devices to libraries designed to support touch-based interactions in Web applications.

In our Smartphones department, Ella Peltonen, Eemil Lagerspetz, Petteri Nurmi, and Sasu Tarkoma discuss their work in understanding and characterizing the energy consumption of a wide variety of system settings available to smartphone users with a technique that leverages crowdsourcing. The authors provide useful tips for readers as well as access to their data, which others can use in their own research.

Our Wearable Computing department focuses on watches. Kent Lyons, in "Smartwatch Innovation: Exploring a Watch-First Model," examines the different types of disruptive innovations and asks whether a smartwatch falls into each type of disruption. He examines whether a smartwatch is an innovation for the smartphone, an innovation for a watch, or a disruptive innovation of its own. He then examines the movement toward "watch first" and what it would take for a watch to be an independent platform of its own. His analysis is well worth reading!

Mary Baker and Justin Manweiller discuss drones, robots, and IoT in Notes from the Community. I haven't followed the developing drones market closely, and I am amazed at the types of drones becoming available. On the robot front, I was surprised to learn that household robots (aka roombas) outnumber industrial robots almost 10 to 1. I particularly liked the discussion of the IoT-enablement of the sushi conveyer-belt restaurants! And the ability to detect depression from a person's smile seems like important progress for diagnosing mental health issues.

Our Innovations in Ubicomp Products column examines how cloud-based AI APIs can be used to enhance your pervasive computing products and prototypes. These APIs include capabilities around speech recognition, computer vision, text analytics, machine learning, and more. Albrecht Schmidt shows readers how to make a sample call to three different AI services and then shows the sample output produced by each. He concludes that incorporating basic AI capabilities into your pervasive applications is relatively simple and requires only minimal programming skills, but that complex capabilities might require someone with expertise in computer vision, speech, or the like, depending on your needs.

In Pervasive Health, Stephen Intille discusses the Precision Medicine Initiative (PMI), which was announced by President Obama approximately one year ago. The PMI, which has also been called personalized medicine, has significant implications for pervasive health research and presents numerous challenges for the pervasive computing community to address in the years to come. Imagine collecting all the health and behavior data for a cohort of more than one million individuals and following those individuals longitudinally for decades. Those of you working in the healthcare space will want to read this article carefully; those of you concerned about your own or your loved ones' health and wellness will find the initiative interesting.

Our upcoming 2016 editorial calendar includes topics that should be of interest to both the pervasive and IoT communities, with special issues on Domestic Pervasive, Pervasive Displays, and Energy Harvesting and Power Management (see the sidebar for more information). I hope that our colleagues in the IoT community will join us in our explorations of these, and future, topics.

REFERENCES

1. M. Satyanarayanan, "A Catalyst for Mobile and Ubiquitous Computing," *IEEE Pervasive Computing*, vol. 1, no. 1, 2002, pp. 2–5.
2. N. Davies, "Twenty Years On," *IEEE Pervasive Computing*, vol. 11, no. 1, 2012, pp. 2–4.
3. K. Ashton, "That 'Internet of Things' Thing," *RFID J.*, 22 June 2009; www.rfidjournal.com/articles/view?4986.
4. G. Press, "A Very Short History of the Internet of Things," *Forbes*, 18 June 2014; www.forbes.com/sites/gilpress/2014/06/18/a-very-short-history-of-the-internet-of-things/2.
5. G. Andrews and H. Hedinger, "The Internet of (Connected) Things," 2014; http://info.neotechnology.com/rs/neotechnology/images/IoT-GraphAlchemist.pdf.
6. C. Floerkemeier and F. Mattern, "From the Internet of Computers to the Internet of Things," Distributed Systems Group, Institute for Pervasive Computing, ETH Zurich, 2010; www.vs.inf.ethz.ch/publ/papers/Internet-of-things.pdf.

Maria R. Ebling is a director at the *IBM T.J. Watson Research Center.* She manages a team building systems capable of supporting a Smarter Planet while not forgetting about the people who use such systems. Ebling received her PhD in computer science from Carnegie Mellon University. She's a member of the IBM Academy of Technology, a distinguished member of the ACM, and a senior member of IEEE. Contact her at ebling@us.ibm.com.

This article originally appeared in IEEE Pervasive Computing, *vol. 15, no. 1, 2016.*

Building Sensor-Based Big Data Cyberinfrastructures


Elisa Bertino
Purdue University


Surya Nepal
CSIRO Data61, Australia


Rajiv Ranjan
Newcastle University

Sensors have been with us for a very long time, but recently they've been used more extensively in almost every industry sector and application—for example, agriculture, manufacturing, healthcare, biophysical and environmental sciences, and smart cities. Two technological trends—Internet of Things (IoT) and cloud computing—are pushing for even wider sensor deployment. IoT is bringing new sensor use cases, while cloud computing is becoming the de facto hosting and processing platform for sensor-generated data. Additionally, advances in nanotechnologies are enabling the deployment of nanosensors, allowing us to collect data at very fine granular scales.

Figure 1 shows a conceptual architecture for a sensor-based big data cyberinfrastructure consisting of three layers.[1] The bottom layer is the sensor network layer, which senses and collects large sets of data in real or semi-real time. Above that is the middleware layer, which includes the engines for processing the collected big data sets. Previous "Blue Skies" departments provided details about the components of this layer.[2,3] The top layer is the application layer, which includes a variety of applications depending on the specific use case and can range from precision agriculture to personalized healthcare.[4,5] Here, we'll concentrate on the middleware tools and technologies needed to facilitate efficient management of interoperations within the sensor network layer (highlighted in green in Figure 1). We'll also highlight some of the research gaps related to, for example, deployment, interoperations, provenance, and data processing of sensor-based cyberinfrastructures.

Challenges

Today we can store and process large-scale datasets in the middleware layer using a mix of software and hardware resources offered by the advancement of cloud and big data technologies. However, these applications' success critically depends on the integration of and access to information collected by sensors. We outline several key challenges in the design and implement next-generation sensor-based big data cyberinfrastructures.

The first challenge is the lack of tools to aid in the design of sensor-based data collection applications. Several key factors need to be considered when designing applications that acquire sensor data. For example, in precision agriculture, unforeseen soil variations (such as fertility, texture, pH, and water status) increase the statistical variance of an individual plant's phenotypic measurements. It's therefore difficult to come up with a sensor placement

 Published by the IEEE Computer Society

strategy to accurately detect meaningful differences among biological treatments for different sensor types.

The second challenge is the lack of tools to aid in the deployment and monitoring of sensors. Continuous monitoring of sensors is essential to prevent data loss. In many cases, sensor data isn't recoverable if the loss occurs because of inefficient caching and communication protocols. If the lost data is critical to understanding the performance of experimental applications (for example, crop yield within a field-based genetic test), the entire experiment can be lost for that year. For this reason, continuous monitoring and early detection and correction of failures are critical.

Another challenge is the heterogeneity in sensor capabilities and wireless link quality. Several issues related to communication protocols need to be addressed. For example, in precision agriculture, current sensors implement a myriad of communication protocols, and the wireless communication links are affected by weather and crop canopies. Another issue related to communication includes the design of energy-efficient protocols because batteries in sensors might not be easily replaceable.

The last challenge we'll address is the lack of provenance and metadata tools for sensors. Provenance is essential for keeping track of which data was collected and/or aggregated for which sensors, and conveying contextual information about sensor deployment and life cycle. Conventional approaches require collection and transmission of large data volumes. We thus need new techniques that reduce the size and enhance the efficiency of provenance and metadata collection, recording, and transmission. Addressing these gaps requires developing cyberinfrastructures to automate the efficient collection, normalization, curation, measurement, and sharing of sensor data. The following sections elaborate on key requirements of such infrastructures as well as approaches and research directions to address some these requirements.

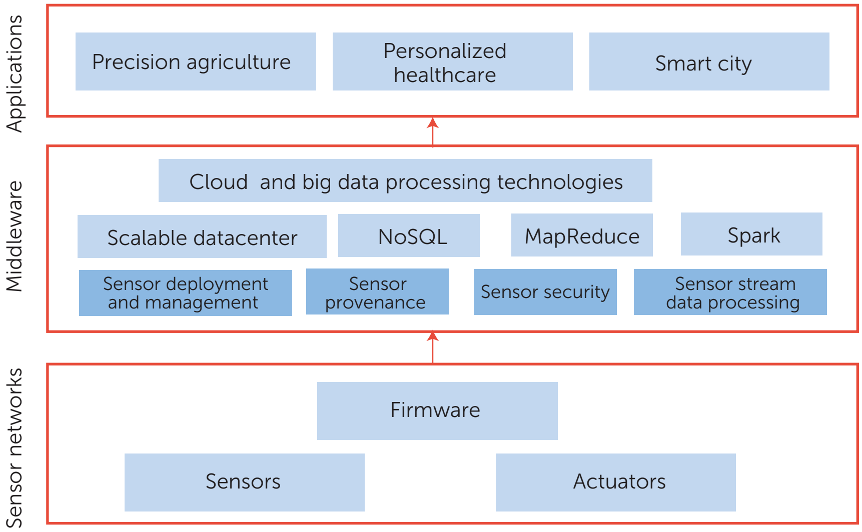


FIGURE 1. Conceptual architecture of sensor-based data-intensive cyberinfrastructures. The architecture contains three layers: sensor network layer, middleware layer and application layer.

Sensor Deployment and Management

First, we need to develop techniques that can support sensor localization, which is achieved through georeferencing sensor data. Supporting tools are therefore needed for the automatic georeferencing of sensors.

Second, we need to develop suitable sensor placement techniques. For example, in precision agriculture, sensors are placed with the soil being measured to generate a soil property map of the entire field. Sensors are used to monitor the growth of crops, and different crops have different sensitivities to soil conditions and are planted at different densities. Therefore, to effectively place sensors, we must consider

- domain-specific knowledge,
- sensor failure and attack probability, and
- wireless link quality.

We also need tools to support optimal application-dependent sensor placement.

Third, we need mechanisms for optimal configuration of the duty cycling and sleep/wake scheduling of sensors. In precision agriculture, during a particular season some sensor data might be important and other data might not be, depending on the crop life cycle. Hence, the cyberinfrastructure needs to include a decision support system that can dynamically and minimally adjust the push/pull rates for each sensor, according to the real-time data processing needs. A reactive programming approach can let programmers define a set of rules for each sensor for how frequently and on which conditions data should be reported to the base station.

Fourth, we need techniques that will ensure robust communications under

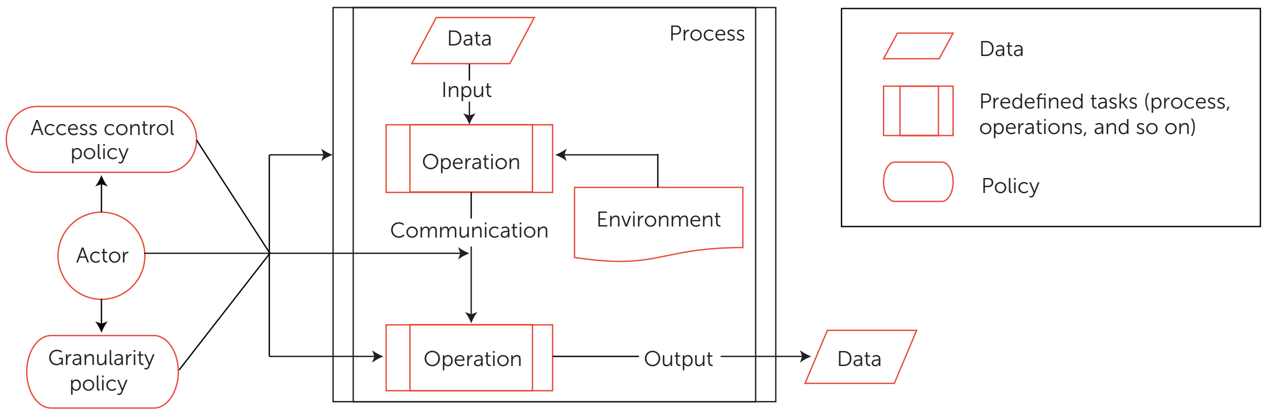


FIGURE 2. Provenance model. The figure provides a graphical representation of a comprehensive provenance model. The key elements of the model are actors that execute operations on data restricted according to the access control policies.

poor link quality. Wireless link quality changes over time due to seasonal changes in foliage size and extent; plant density variations; weather, wind, and moisture; and seasonal changes in crop canopy height and density. This can result in reliability degradation, redundant transmissions, and increased contention among node transmissions. We can exploit link correlation to eliminate the overhead of explicit control packets in networks with high correlation, and use network coding to pipeline transmission of multiple packets, attempting to utilize only a single timer per node for increased scalability.

Finally, we need mechanisms for continuously monitoring sensors to verify that they work as expected. Sensor monitoring is a complex task because multiple factors—calibration errors, environmental conditions, attacks, decay of sensor energy, and so on—can lead to sensor malfunctions. Different monitoring techniques might thus have to be combined and deployed, ranging from simple, such as profiling sensor baseline behavior, to complex, such as fine-grained diagnosis techniques for sensors.[6]

Sensor Provenance

Provenance of sensor data is critical in many applications because it

- plays a key role in assessing and ensuring data trustworthiness;
- aids in preventing data losses; and
- ensures the repeatability of scientific experiments and processes.

Provenance must be fine grained to be able to track which sensors acquired and/or transmitted specific data items. Provenance information should include location and time of the data acquisition as well as all processing steps executed on data, such as conversion, integration, and aggregation. These steps can be executed on different subsystems within the middleware, including sensor deployment and management systems, relational database management systems, stream-processing systems, distributed in-memory data stores, and NoSQL databases. The hard research challenge is to capture such provenance operations across multiple, heterogeneous big data processing technologies and subsystems. Finally, provenance must be secured to prevent provenance information from being tampered with or lost.

Provenance Model

An important step in the development of tools for managing provenance information is developing an expressive provenance model capable of representing the provenance of data objects with various semantics and granularity. Such a model should be able to capture data provenance in a structured way as well as to encapsulate the knowledge of both the application semantics and the system. At the same time, it should support provenance interoperability. Figure 2 shows one such model.[7] This model addresses various requirements, including expressive power, flexible level of provenance granularity, and security, and is compatible with existing provenance standards.[7]

In this model, data creation or manipulation is performed by a sequence of operations initiated by a process.[7] A process can be a service or activity in a workflow, a user application, or an operating-system-level process (for example, Unix). An operation executes specific tasks and causes manipulation of some system or user data. Thus, the operations generate or modify not only persistent data but also intermediate results or system configurations. Communication represents the interaction (such as dataflow) between two processes or between two operations in a process. Web services, user applications, and Unix processes are examples of processes; statements within an executable function or command line exemplify the operations; and dataflow, copy/paste, interprocess communication in Unix, and so on, represent the communication between operations or processes. An operation might take data as input and output some data. Each data object is associated with a lineage record that specifies the immediate data objects used to generate the data. Lineage is particularly helpful for producing a data object's data dependency graph. Processes, operations, and communications are operated by actors, which can be human users, workflow templates, and so on. When data provenance is used to detect intrusion or system changes, knowledge of a user role or the workflow template might be helpful. Environment refers to the operational state, parameters, and system configurations that also affect an operation's execution and thus the output data. This additional provenance information is crucial for understanding the performance of the operation and the nature of the output. In addition, the model supports the specification of granularity policies, allowing users to specify the level of detail for the provenance information to be captured and stored. It also includes, as part of provenance, information on the access control policies under which data have been accessed, which is crucial when investigating data breaches.

A relevant open research direction concerning the use of such a model within cyberinfrastructures for sensor management is the definition of services to be associated with the provenance data. To effectively use the provenance information, the model must support various types of provenance queries (on entity attributes, invocations, lineage, provenance, and so on). Historical dependencies as well as subsequent uses of a data object should be easy to track. If a data object is processed across multiple big data processing technologies and subsystems (see Figure 1), an administrator might want to see a high-level machine, system, or domain view of the provenance graph. In addition, to find relevant information from large provenance graphs, one should be able to filter, group, or summarize all or portions of the provenance graphs and generate tailored provenance views. Thus, the model should be able to distinguish the provenance generated from different systems and construct specialized views of provenance graphs.

When data provenance is used to detect intrusion or system changes, knowledge of a user role or the workflow template might be helpful.

Provenance Transmission

In a multihop sensor network, data provenance allows a base station to trace the source and forwarding path of an individual data packet from its generation. However, the limited resources and bandwidth constraints of the sensor nodes make collecting data provenance for each packet challenging. Lightweight data provenance schemes have been developed to address these limitations.[8,9] In these schemes, data provenance is represented as a directed graph, where each vertex represents node's provenance record and each edge the direction of data transmissions between nodes. One major issue with these schemes is that the provenance size increases with the number of nodes in the dataflow path. Therefore, we need lightweight provenance solutions for each data packet that don't introduce significant overhead.

A more recent approach uses arithmetic coding to compress the dataflow path.[10] Arithmetic coding is a lossless data compression technique that assigns short code words to more probable events and longer code words to less probable events.[11] In such a scheme, each sensor represents an event and its occurrence probability is used for provenance encoding and decoding. Such schemes achieve a provenance compression ratio that approaches Shannon's theoretical entropy bound and thus outperforms the better-known Huffman-coding-based compression schemes. However, the deployment of such a scheme in sensor networks requires addressing several additional issues, such as handling dynamic wireless sensor networks, handling cases in which packets are routed in more than one node at a time, and providing secure sensor location information.

Sensor Security

Many scenarios of interest to sensor-based applications require assuring data integrity and confidentiality as well as authentication among different sensors and other parties. Sensors are vulnerable to malicious attacks such as impersonation, interception, capture, or physical destruction, due to their unattended operative environments and lapses of connectivity.[12] To address security issues, encryption key management protocols need to be deployed. Current approaches, such as those based on symmetric key encryption, elliptic curve public key cryptography, and identity-based public key cryptography, suffer from one or more of the following drawbacks:

- high communication and storage overhead,
- vulnerability to impersonation and key compromise attacks,
- certificate management overhead and computational overhead for pairing operations, and
- inability to support mobility.

A novel pairing-free certificateless public key cryptography-based key management (CL-EKM) scheme for dynamic wireless sensor networks (WSNs) addresses the drawbacks of previous approaches.[13] In certificateless public key cryptography (CL-PKC),[14] a party's full private key is a combination of a partial private key generated by a key generation center (KGC) and the party's own secret value. The CL-EKM scheme supports the establishment of four types of keys for each sensor node:

- a certificateless public/private key pair for the sensor,
- an individual key shared with the base station,
- a set of pairwise keys for each sensor node in the same sensor cluster,
- a cluster key shared among all the neighboring nodes.

However, we still need extensive experimental evaluations using different types of sensors and networks as well as encryption schemes that support data aggregation and processing at the sensor network level. Techniques for securing the software running on the sensors[15] and for quickly responding to security incidents[16] are also needed.

Sensor Streamed Data Processing Techniques

Two main strategies can be devised for processing the data acquired from sensors.

Data Warehousing

This approach involves shipping raw sensor readings from the sensor network to a central repository stored in a cloud for subsequent analysis. It's most appropriate when all data needs to be kept and there's sufficient energy to transmit the data outside the sensor network or collection. A number of open source data streaming tools are available to be used in this approach. These include Apache Samza, Esper, Spark Streaming, Apache Storm, and Apache S4.[3]

In-Sensor Network Processing

This approach involves executing certain data operations in the sensor network itself and shipping outside the results of these operations. Examples include computing some simple aggregated functions, such as computing the maximum among a set of sensor readings and performing outlier detection. This approach is most appropriate in cases where energy saving can be gained by locally processing the data and data quality can be increased by locally implementing data cleansing actions. Clearly, both approaches need to be supported by cyberinfrastructures used depending on the requirements of specific applications. Approaches like Cougar, which provides in-network query processing, are needed for processing streaming data.[17,18] Fog computing, another emerging technology that has potential to support in-sensor-network processing,[20] is a virtualized platform located at the edge of the network that provides compute, storage, and network services between the end devices and the cloud datacenters. Its characteristics such as low latency, location awareness, support for large number of nodes, and widespread geographical distribution make it suitable for streaming and real-time applications, and thus represents a potential solution for in-network processing for sensor-based applications.[19]

Cloud technology provides a foundation for emerging data-intensive applications. Cloud-based infrastructure services support the data life cycle from ingestion to analysis to storage. However, building cyberinfrastructure for such applications is still challenging. Here, we've discussed a few recent approaches and outlined open research directions. We believe that current technology trends in IoT and embedded systems will further push the need for cloud-based cyberinfrastructures for sensor-based applications.

References

1. C. Perera et al., "Context Aware Computing for the Internet of Things: A Survey," *IEEE Comm. Surveys & Tutorials*, vol. 16, no. 1, 2014, pp. 414–454.

This article originally appeared in IEEE Cloud Computing*, vol. 2, no. 5, 2015.*

2. L. Wang and R. Ranjan, "Processing Distributed Internet of Things Data in Clouds," *IEEE Cloud Computing*, vol. 2, no. 1, 2015, pp. 76–80.
3. R. Ranjan, "Streaming Big Data Processing in Datacenter Clouds," *IEEE Cloud Computing*, vol. 1, no. 1, 2014, pp. 78–80.
4. A. Baggio, "Wireless Sensor Networks in Precision Agriculture," *ACM Workshop Real-World Wireless Sensor Networks* (RealWSN 05), 2005.
5. S. Coyle et al., "BIOTEX—Biosensing Textiles for Personalised Healthcare Management," *IEEE Trans. Information Technology in Biomedicine*, vol. 14, no. 2, 2010, pp. 364–370.
6. B. Shebaro, D. Midi, and E. Bertino, "Fine-Grained Analysis of Packet Losses in Wireless Sensor Networks," *Proc. 11th Ann. IEEE Int'l Conf. Sensing, Comm., and Networking* (SECON 14), 2014.
7. S. Sultana and E. Bertino, "A Comprehensive Model for Provenance," *Advances in Conceptual Modeling*, LNCS 7518, Springer, 2012, pp. 121–130.
8. B. Shebaro et al., "Demonstrating a Lightweight Data Provenance for Sensor Networks," *Proc. ACM Conf. Computer and Comm. Security* (CCS 12), 2012, pp. 320–328.
9. S. Sultana et al., "A Lightweight Secure Scheme for Detecting Provenance Forgery and Packet Drop Attacks in Wireless Sensor Networks," *IEEE Trans. Dependable and Secure Computing*, vol. 12, no. 3, 2015, pp. 256–269.
10. S.R. Hussain et al., "Secure Data Provenance Compression Using Arithmetic Coding in Wireless Sensor Networks," *Proc. IEEE 33rd Int'l Performance Computing and Comm. Conf.* (IPCCC 14), 2014, pp 1–10.
11. J.S. Vitter and P.G. Howard, "Arithmetic Coding for Data Compression," *Information Processing and Management*, 1994, pp. 749–763.
12. M.A. Rassam, M.A. Maarof, and A. Zainal, "A Survey of Intrusion Detection Schemes in Wireless Sensor Networks," *Am. J. Applied Sciences*, vol. 9, no. 10, 2012, pp. 1636–1652.
13. S.H. Seo et al., "Effective Key Management in Dynamic Wireless Sensor Networks," *IEEE Trans. Information Forensics and Security*, vol. 10, no. 2, 2015, pp. 371–383.
14. S. Al-Riyami and K. Paterson, "Certificateless Public Key Cryptography," *Advances in Cryptography* (ASIACRYPT 03), vol. 2894, 2003, pp. 452–473.
15. J. Habibi et al., "DisARM: Mitigating Buffer Overflow Attacks on Embedded Devices," *Proc. 9th Int'l Conf. Network and System Security* (NSS 15), 2015.
16. S. Sultana, D. Midi, and E. Bertino, "Kinesis: A Security Incident Response and Prevention System for Wireless Sensor Networks," *Proc. 12th ACM Conf. Embedded Network Sensor Systems* (SenSys 14), 2014, pp. 148–162.
17. Y. Yao and J. Gehrke, "The Cougar Approach to In-Network Query Processing in Sensor Networks," *ACM SIGMOD Record*, vol. 31, no. 3, 2002, pp. 9–18.
18. U. Srivastava, K. Munagala, and J. Widom, "Operator Placement for In-Network Stream Query Processing," *Proc. 24th ACM SIGMOD-SIGACT-SIGART Symp. Principles of Database Systems* (PODS 05), 2005, pp. 250–258.
19. F. Bonomi et al., "Fog Computing and Its Role in the Internet of Things," *Proc. 1st Edition of the MCC Workshop on Mobile Cloud Computing*, 2012, pp. 13–16.
20. L.M. Vaquero and L. Rodero-Merino, "Finding Your Way in the Fog: Towards a Comprehensive Definition of Fog Computing," *SIGComm Computing Comm. Rev.*, vol. 44, no. 5, 2014, pp. 27–32.

ELISA BERTINO *is professor of computer science at Purdue University and director of the Purdue Cyber Center. Her research interests include data security and privacy, security of embedded systems and sensor networks, and cloud security. Bertino received the doctorate in computer science from the University of Pisa (Italy). She's a fellow of IEEE and of ACM. Contact her at bertino@purdue.edu.*

SURYA NEPAL *is a principal research scientist in the Data61 Business Unit at the Commonwealth Scientific and Industrial Research Organization (CSIRO), Australia. His research interests include cloud computing, big data, and cybersecurity. Nepal has a PhD in computer science from Royal Melbourne Institute of Technology, Australia. Contact him at surya.nepal@csiro.au.*

RAJIV RANJAN *is an associate professor in the School of Computing Science at Newcastle University, UK, and a visiting scientist at the Commonwealth Scientific and Industrial Research Organization (CSIRO), Australia. His research interests include cloud computing, content delivery networks, and big data analytics for Internet of Things (IoT) and multimedia applications. Ranjan has a PhD in computer science and software engineering from the University of Melbourne. Contact him at raj.ranjan@ncl.ac.uk or http://rajivranjan.net.*

Taking the Internet to the Next Physical Level

Vint Cerf and Max Senges, Google Research

Our physical universe has been transformed by computing's ubiquity. The authors describe the challenges and delights we'll find in a future enabled by the Internet of Things.

FROM THE EDITOR

With the realization of the ideas behind the Internet of Things (IoT)—a network of everyday items with embedded computers that can connect directly or indirectly to the Internet—we're entering the era of ubiquitous computing. As the IoT takes root, the number of devices connecting to the Internet is likely to increase 10- or even 100-fold over the next 10 years, forever changing our relationship with "things"—now they'll be smart: smart devices, smart homes, smart buildings, and smart cities.

Although its origins date back to 1999, the IoT's core ideas were first described in Mark Weiser's vision of ubiquitous computing in 1988. Although these ideas have been around for more than 25 years, it has only recently become practical for high-performance processing and networking to be built into everyday products. We now have the ability to augment our things' capabilities at a reasonable cost and size: this embedded computing—with the equivalent performance of a complete 1980s-era workstation—can be added to products for less than $10.

To kick off the inaugural installment of "The IoT Connection," a bimonthly forum bringing *Computer* readers exciting developments from the IoT field, it seems appropriate to invite one of the fathers of the Internet, Vint Cerf, and one of Google's in-house philosophers, Max Senges, to get the ball rolling. Vint's perspective, spanning his considerable experience in networking from the early days of Internet design at DARPA to its modern instantiation, is combined with Max's expertise in building both sociotechnological innovation around a "good IoT" and a vibrant multistakeholder IoT ecosystem.

Please contact me with your content suggestions, especially regarding IoT standards development, applications, protocols, security and privacy, and novel human–computer interaction requirements for new modes of use. —*Roy Want*

We've come a long way since the article in which Mark Weiser envisioned small, ubiquitous, connected computers that enhanced all aspects of our lives.[1] Here, we present our analysis of the architectural leitmotifs that should be pursued so the Internet of Things (IoT) ecosystem can enjoy the staggering success of the Internet, which resulted in the World Wide Web. By success, we mean the economic value and the social and technological innovation these platforms have brought to the world.

THE IOT IS HERE

As with the Internet, it's difficult to pin down the dimensions of value creation through the IoT because it's essentially a general-purpose platform. So, we'll start by highlighting some examples of how the IoT already helps society in many different ways, through applications ranging in scope from the individual to the

 2469-7087/16/$33.00

EDITOR ROY WANT
Google; roywant@google.com

Figure 1. The scope of the Internet of Things (IoT). (Source: Copenhagen Wheel [http://senseable.mit.edu/copenhagenwheel] photo by Max Tomasinelli; www.maxtomasinelli.com. Aclima photo courtesy of Aclima [http://aclima.io].)

planetary (as shown in Figure 1), as well as across ventures in a variety of industries.

IoT ventures are rooted in and advance all kinds of professional spheres, including entertainment (for example, mixed-reality ventures like Magic Leap; www.magicleap.com), science (such as scientific data sharing), education (for example, connected platforms like SAM Labs; http://samlabs.me), health (such as the smart contact lens developed at Google X), and civic innovation (for example, participatory smart city initiatives; https://smartcitizen.me).

UNDERSTANDING THE CHALLENGES

Although we're already reaping so many of the IoT's exciting benefits and anticipating much more from the promising forecasts of its future, mainstream users and organizations aren't yet craving the majority of IoT devices and services. Indeed, some of its potential applications and complexities stir public fears over privacy and security risks—an aspect the media tends to revel in reporting about. Additionally, many IoT products have a level of complexity that limits their appeal for users unwilling to invest time and resources to learn to configure them.

However, its greatest limitation is arguably the lack of open standards, because the IoT's growth will bring many incompatible IoT solutions. Even if standards are used, consumers are hesitant to pay a premium for IoT-enabled devices, particularly if these devices aren't compatible with products and devices they already own. As many IoT products—such as home appliances and cars—have a product lifetime of more than a decade, consumers need to be confident that technical support and security updates are available long term.

LESS IS MORE

Tech companies have generally pursued business models in which successful products and services are constantly updated. This translates to technology-rich environments where devices and services constantly compete for consumers' attention; thus, technology tends to distract rather than add value. Will the addition of more networked devices add more screens to this cacophony? We have no doubt that IoT technology will profoundly impact our lives. If these devices are to fit into our lives comfortably, they shouldn't require more screens or keyboards.

Figure 2 illustrates what our colleagues at Nest Labs—a home automation producer of programmable, Internet-connected thermostats, smoke detectors, and security systems—came up with when their CEO, Tony Fadell, asked them to envision the living room of the future. What's important is

Figure 2. Nest's vision of the living room of the future. (Source: Nest Labs)

"IoT is only a bunch of connected gadgets."	**"IoT objects are embedded in the fabric of my life."**
"It's not secure."	"I'm in the center of my IOT ecosystem and I'm in control of my data and its privacy."
"It's just a gimmick; it won't last."	"It's useful on many levels. I can see it's here to stay and I'm happy about that."
"It's complicated."	"It's easy and built to make complex things become easier."
"I have no idea what I'm getting myself into."	"I understand what I'm doing and the tradeoffs I'm assuming."
"It's a configuration nightmare."	"It was so easy, most devices configured themselves."

Figure 3. From resisting to embracing the IoT. Our assessment of current user perception of the IoT and where we believe the value propositions should be.

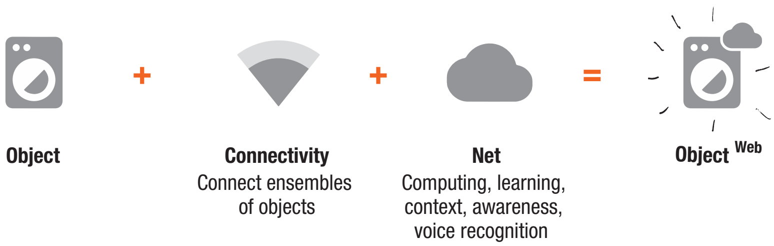


Figure 4. How connectivity changes "things."

what's missing—where's the collection of screens and keyboards? We might interact with future devices through voice requests, gestures, or perhaps inputs mediated through smartphones.

As Weiser espoused, technology will (or should) fade into the background, supporting us in our private and professional lives in many subtle and effective ways.

THE PROMISE OF GOOD IOT TECHNOLOGY

How can we guide technologists, entrepreneurs, and user-experience designers to shift their perspectives? Let's start by comparing current user perceptions with the experiences we'd like to provide (see Figure 3).

When thinking about the IoT, we like the dualism of *hard IoT* versus *soft IoT*—an idea put forward by Usman Haque in 2002.[2] Hard IoT is traditionally understood as a network of electronic gadgets, software, and sensors that are connected so objects can collect and exchange data. In contrast, soft IoT focuses on the value that can be derived from the collection of fluid relationships among people, objects, and spaces.

The following three maxims can inform good IoT design:

- reimagine ordinary objects with the power of the Internet,
- foster ensembles of objects and services, and
- match relevant objects and services for genuine user benefit.

Reimagining ordinary objects with the power of the Internet

How useful will objects be when they're amplified by everything the Internet can do? Figure 4 shows how a traditional "offline" object is enhanced by being connected to the online ecosystem.

Imagine a washing machine with the power of the Web (see Figure 5). Whereas traditional washing machines have all the features they'll ever have once they're installed in a home—they're

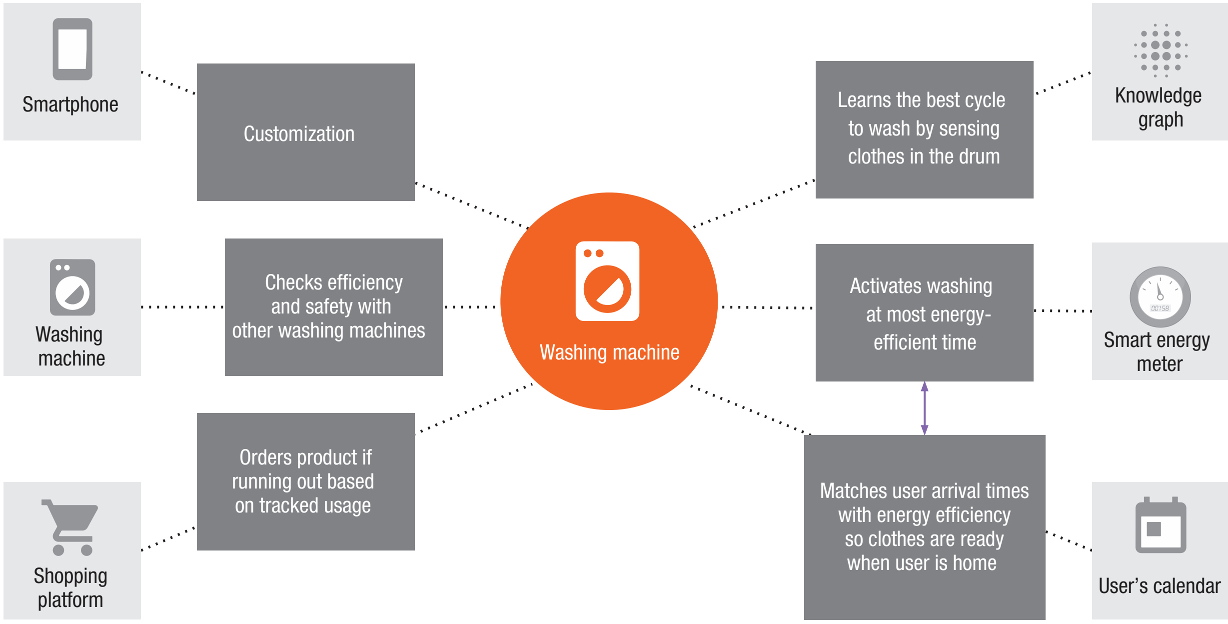


Figure 5. The various features and benefits enabled by connectivity and access to an online ecosystem.

neither customizable nor aware of the resources they consume—a Web-enabled machine is able to acquire new safety or security features and services (for example, programs designed to clean innovative textiles, such as Google's Project Jacquard [http://levistrauss.com/unzipped-blog/2015/05/google-levis-project-jacquard]). Connecting to cloud services and the Internet ecosystem enables the washing machine, for example, to link to other objects and build ensembles that complement and cooperate with one another.

In such configurations, Web-enabled appliances benefit from continuous machine learning, improving their understanding and consideration of context and enabling their access to ecosystem services (such as weather data), markets (such as placing an order to replenish washing detergent; www.youtube.com/watch?v=U1XOPIqyP7A [at 3:02]), and APIs and protocols that allow full cooperation with other devices. Thus, network-enabled objects embody smart behaviors that make them adaptive to new circumstances, more resource efficient, and generally user optimized.

Fostering ensembles of objects and services

Orchestration among an ensemble of objects can add to their usefulness and value—it facilitates rich, intuitively interactive or standardized environments. In the washing machine example shown in Figure 5, users can customize and save personal preferences via their smartphone. The machine can compare efficiency and learn about safety hazards from other machines, track usage and order replenishment of supplies, access knowledge bases to learn the most suitable programs for washing clothes, and find the best price for energy.

Although this is a very simplistic example, such IoT innovations can allow for resource conservation and energy efficiency in a scalable way. It follows that such efficiencies can be realized at a much larger scale, and in industrial IoT ecosystems that can result in huge cost reductions along with green efficiency benefits.

Matching relevant objects and services for genuine user benefit

So far we've considered mainly static actors, but the IoT's true benefits will be felt when we connect contextually relevant objects to the right information and services.

Figure 6 shows how the IoT extends the information graph created by the Web, the social graph created by user-generated media, and the physical graph that links objects and their functionalities. Only when these three knowledge domains are combined can products and services be truly "smart."

PREREQUISITES FOR A GOOD IOT

There are already many successful IoT products and services, and even some (limited) domain-specific ecosystems on the market. Nevertheless, we identify three areas that require significant R&D investment and cooperation before an ecosystem can emerge to universally interconnect all industries, people, and spaces (see Figure 7):

- data, access-control, and identity management;
- standardized and modular system architecture (including protocols and IoT schema); and
- new human–device interaction paradigms and techniques.

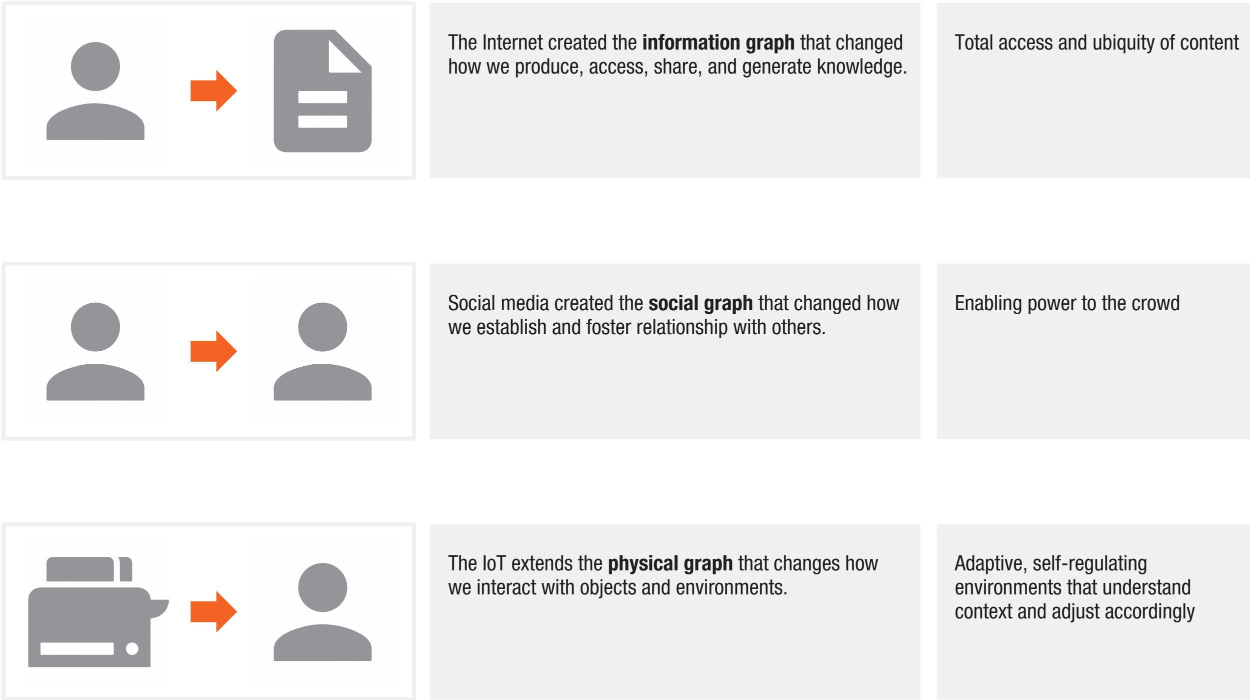


Figure 6. How the IoT extends the Web-created information graph, the user-generated media–based social graph, and the object–functionality physical graph to create "smart" ecosystems.

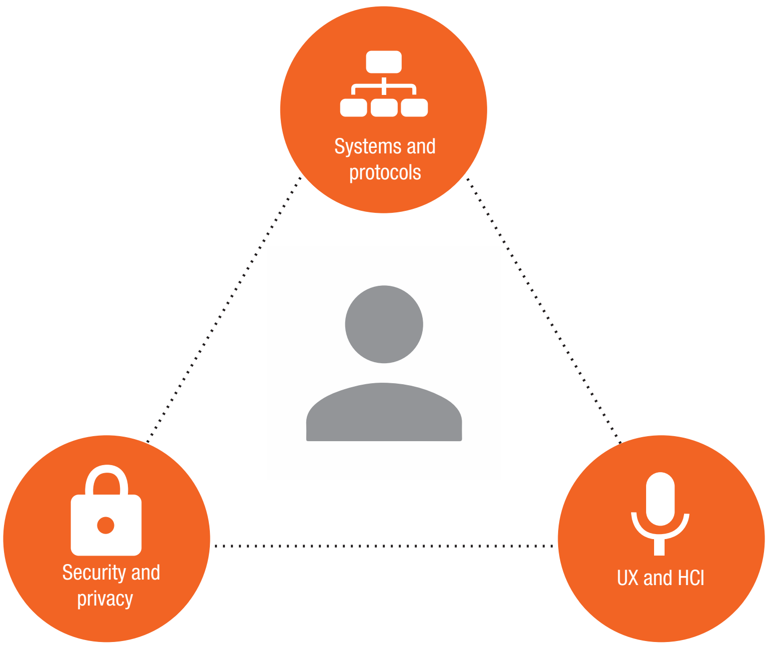


Figure 7. Components of an IoT ecosystem. HCI: human–computer interaction; UX: user experience.

Importantly, in our assessment, all three areas are more likely to result from open peer-production scenarios as described by Yochai Benkler.[3] This will lead to better standards that make it feasible for users to learn nuanced common practices that can be applied internationally across companies, product categories, and industrial and consumer ecosystems. For example, controlling devices with gestures or managing complex dataflows are two new areas where open standards would be beneficial.

Data, access-control, and identity management

In the Internet's early development, user privacy and identity management—and security, to some extent—weren't at the forefront of its inventors' minds, and were only incorporated into the service network much later. User safety has also recently become deeply relevant as devices like cars and door locks become networked.

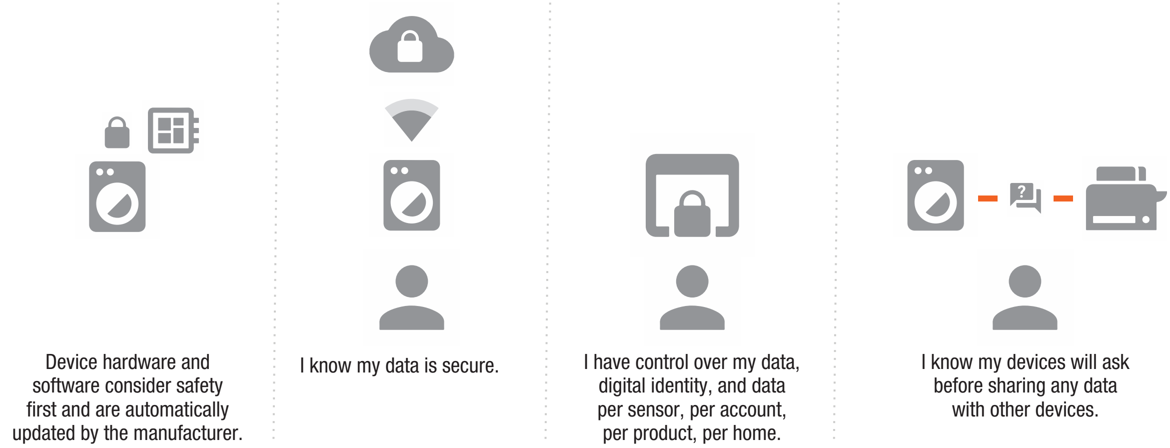


Figure 8. Embedding trust into the IoT.

We believe that a system architected from the ground up, with identity management and data ownership as core features, will better serve users in a world of networked things.

To do this, we'll first need a solid identity management system. All IoT objects—such as door locks and cars—must have deeply ingrained, authority-based usage rights. Establishing preferred usage patterns (personalization) is fundamental, especially to reap the benefits of ensembles and spaces (in other words, to avoid constantly configuring and adapting settings). Because everything creates data, we'll also need to clearly define its flow and ownership to allow for reasonable and effective storage and management.

The key to mainstream IoT acceptance lies in a decentralized, user-controlled system with strong data management and identity controls to elicit greater trust and adequate privacy. Security and safety can be handled mostly by service providers. Figure 8 illustrates the main requirements of a trust-generating identity- and data-management system.

Standardized and modular system architecture

The current IoT landscape is made up of individual solutions, or "walled gardens," that offer special perks for customers who buy from the "product family." Although the Internet was developed around open standards, AOL's and CompuServe's walled gardens were among the first of the Internet's initial development and deployment experiments. We now know that the open ISP model provided superior services to customers, but this experiment needs to be repeated at the beginning of the IoT era.

We're not dogmatic about openness, but it seems clear to us that the Internet's success is based on the level playing field created by open standards and interoperability. A successful IoT ecosystem will allow start-ups, established small and midsize businesses (SMBs), and big companies to plug in and play a role in building viable products. For the IoT to become a mainstream success, the IKEAs, Holiday Inns, and Disneys of the world—along with all kinds of SMBs—must join the party and help foment ever more connected hardware and services.

Nearly 100 IoT consortia and standardization efforts are underway. We appreciate the competition to create the best system, but from a strategic perspective, it's more desirable for the architecture of platforms, schemata, and protocols to be modular and for their core elements to be maintained by transparent meritocratic organizations dedicated to the public interest.

Particularly problematic for the mainstreaming of IoT products and services is the "app trap": the tendency for each connected thing to develop and require its own smartphone application. We need to move away from this paradigm. Single objects or systems shouldn't rely on smartphones as controllers, but should instead use common APIs so that various devices and programs can access and control them (given the right credentials). This will enable an ecosystem in which users interact with multiple devices through voice, gestural, and touch interfaces as devices share contextual information. Standardization will also improve efficiency by enabling competition and user choice for managing and controlling ensembles of devices.

New paradigms for user interface and interaction design

Last but not least, interacting with our connected "things" shouldn't revolve around putting little touchscreens on all of them. Keyboards and mice aren't an effective means to use and orchestrate the devices surrounding us. As neither of us is an expert in human–computer interaction or user-experience design,

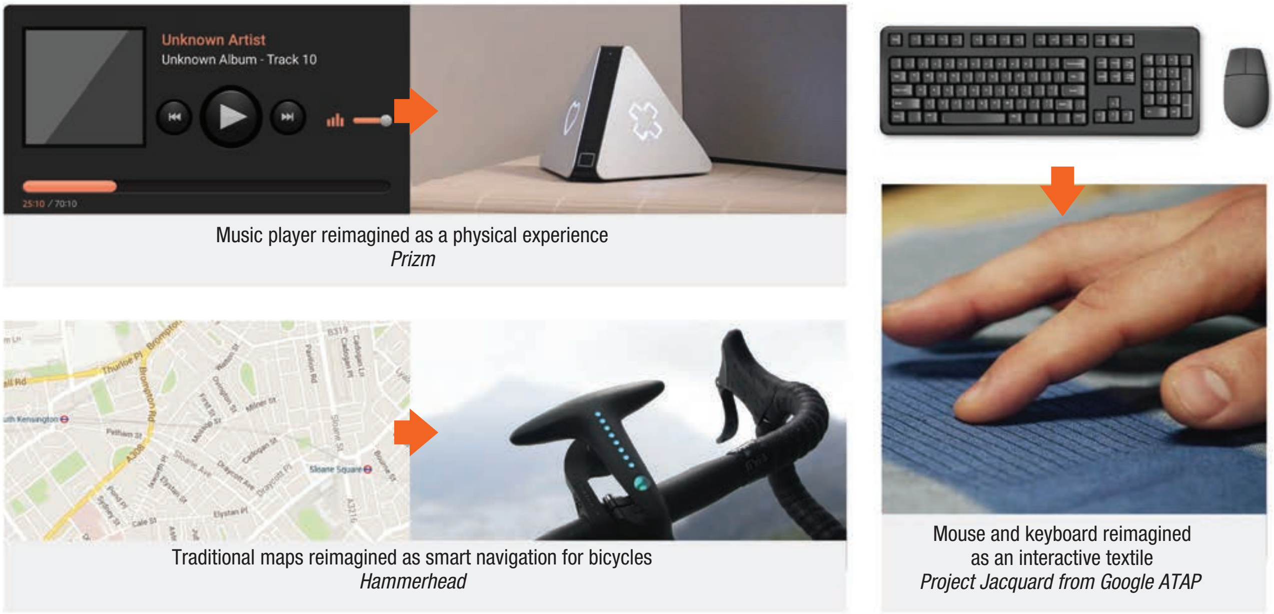


Figure 9. Reimagining interaction with "things" beyond traditional interfaces.

we refrain from making assessments about these challenges and opportunities. Instead, we provide a few examples of nontraditional IoT interfaces in Figure 9.

It's not possible in such a short article to comprehensively analyze the IoT with its multifaceted dimensions. We hope, however, that our analysis here might illustrate the IoT's core potential and articulate some of the barriers to the adoption of a universal, mainstream IoT. As we strongly believe in openness and collaboration, we very much look forward to teaming up and building a good IoT with all of you.

Let's expand the Internet's success story in terms of permissionless innovation and level the playing field for all competing innovators.

Let's promote an interoperable ecosystem based on open standards.

Let's make identity, access-control, and data management an essential part of the technological architecture from the start of the IoT evolution.

Let's take the Internet to the next physical level. ■

REFERENCES

1. M. Weiser, "Hot Topics: Ubiquitous Computing," *Computer*, vol. 26, no. 10, 1993, pp. 71–72.
2. U. Haque, *Hardspace, Softspace and the Possibilities of Open Source Architecture*, white paper, Haque Design + Research, 2002; www.haque.co.uk/papers/hardsp-softsp-open-so-arch.PDF.
3. Y. Benkler, *The Wealth of Networks: How Social Production Transforms Markets and Freedom*, Yale Univ. Press, 2006.

NOTE

This article is an elaboration on Vint Cerf's keynote address at the IEEE 2nd World Forum on Internet of Things (WF-IoT 15) in Milan, Italy, in December 2015.

VINT CERF is vice president and Chief Internet Evangelist at Google Research.

MAX SENGES is a program manager at Google Research. Contact him at maxsenges@google.com.

This article originally appeared in Computer, *vol. 49, no. 2, 2016.*

COMPUTER ENTREPRENEUR AWARD

In 1982, on the occasion of its thirtieth anniversary, the IEEE Computer Society established the Computer Entrepreneur Award to recognize and honor the technical managers and entrepreneurial leaders who are responsible for the growth of some segment of the computer industry. The efforts must have taken place over fifteen years earlier, and the industry effects must be generally and openly visible.

The award is a museum-quality sterling silver chalice commissioned from Washington DC artist Michael Schwartz. The gold-plated crown below the cup displays the alternating symbols of the IEEE and the Computer Society, supported by clusters of laurel leaves, an ancient symbol for outstanding achievement.

All members of the profession are invited to nominate a colleague who they consider most eligible to be considered for this award. Awarded to individuals whose entrepreneurial leadership is responsible for the growth of some segment of the computer industry.

DEADLINE FOR 2017 AWARD NOMINATIONS

DUE: 15 OCTOBER 2016

- The award nomination requires a minimum of 3 endorsements
- Presented at the highly acclaimed IEEE-CS Annual Awards Ceremony.

Past Recipients Include: Diane Greene, Mendel Rosenblum, Sandy Bosack, John Warnock, Charles Geschke, Michael Dell.

AWARD SITE: https://www.computer.org/web/awards/entrepreneur

www.computer.org/awards

IEEE · IEEE computer society

Elastic Stream Processing for Distributed Environments

Christoph Hochreiner, Stefan Schulte, and Schahram Dustdar • *TU Wien, Austria*

Freddy Lecue • *IBM Research–Ireland*

The Internet of Things introduces the need for more flexibility in stream processing. To address these challenges, the authors propose elastic stream processing for distributed environments. This novel concept allows for scalable and more flexible solutions compared to traditional approaches.

Big Data has become an immanently important topic in many application areas such as smart cities, smart factories, the smart grid, or smart mobility. Gartner estimates that there are already 1.1 billion connected physical devices alone for smart cities in 2015, and expects this number to rise to almost 10 billion devices in 2020 (www.gartner.com/newsroom/id/3008917). These connected devices form the so-called *Internet of Things* (IoT).

Although the IoT enables new opportunities in different industries and application areas, its potential benefits are only realizable once important research questions in the field of Big Data have been answered. These questions range from data access, storage, discovery, analysis, or processing to reasoning on Big Data. Especially *smart systems* — such as smart cities and the IoT — require real-time processing capabilities (including stream processing) to provide value-added services.[1] Although stream processing has been an active research area for several years, the topic is gaining momentum as the emerging IoT increases the amount of continuous streaming data (such as sensor data) to previously unknown levels.

IoT-based value-added services are triggered by sensor events or user requests, which lead to variable system loads being handled by stream processing systems. To cope with these challenges, a stream processing system must be elastic in terms of its processing capabilities.

Regarding elasticity, we consider three interdependent dimensions. The first dimension is *quality elasticity*. This deals with aspects such as the response time, the quality of a response for queries against streaming data, or the amount of successfully processed streaming data. If too much data must be handled by a stream processing system, this can lead to a reduction in service quality[2] or to all non-processable data being dropped.

The second dimension is *resource elasticity*. It indicates that the amount of computational resources required to process streaming data must be adapted at runtime to cope with the current system load.[3] Until now, most stream processing systems ran on fixed computational resources in one location, such as a data center, and could distribute the workload only among these fixed resources. In fact, systems with fixed resources could adapt to volatile data rates only by reducing the quality of service (QoS).

The third dimension is *data elasticity*. This means that data aren't necessarily stored or processed locally, but rather in various remote data repositories in the Cloud. NoSQL approaches, such as MongoDB (www.mongodb.org), already envision the distributed storage of huge amounts of data, but until now, stream processing hasn't applied these NoSQL concepts.

To take stream processing to the next level, resource elasticity is a prerequisite to maintain service-level agreements (SLAs), regardless of the system load. It's also an efficient approach to elastically deal with huge amounts of stored data located in different geographical locations. In order to back up our claims, we provide a scenario from the mobility domain (see the related sidebar), propose elastic stream processing for distributed environments, and identify several open research challenges. Let's begin by taking a closer look at the proposed model.

Conceptual Overview

To cope with the changing resource requirements and different geographical locations of sensors as well as the stream processing nodes, we propose an elastic stream processing model for distributed environments. Our model is composed of multiple self-contained nodes (see Figure 1).

As we discuss in the sidebar "Related Work in Stream Processing," in the past stream processing systems generally pursued a monolithic approach where a centralized component manages the communication and execution of stream processing operations. These operations range from filtering data, to querying sensor data for a specific time span, to customizing business logic (for example, transforming sensor data into other data formats).

Monolithic systems are deployed in one geographic location (on a cluster, for example), which eases the communication among the different processing nodes as well as the resource management for the processing operations. In the past, static stream processing scenarios with short and reliable communication paths successfully applied this centralized approach. Nevertheless, this approach is unsuitable for the IoT, because IoT scenarios are likely to evolve over time, in the sense that a smart city is constantly evolving. This leads to continuous changes of the IoT landscape, and therefore also to the stream processing infrastructure. To counter the IoT domain's challenges, we must design an elastic stream processing model consisting of self-contained nodes that we can orchestrate during runtime.

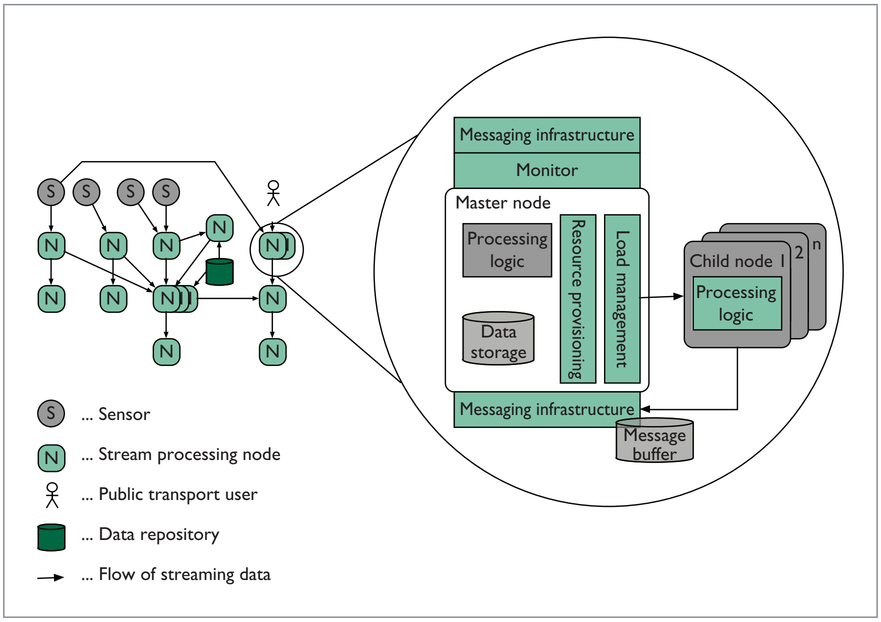


Figure 1. Elastic stream processing model. On the left we show the choreography based on a motivational scenario, and on the right we show a processing node in detail.

The left side of Figure 1 shows an exemplary choreography based on a motivational scenario, and the right side presents one of the processing nodes in detail. Such a node is deployed on fixed computational resources (for instance, in a cloud-based infrastructure), where each node can autonomously allocate computational resources according to the data rates. To realize elastic scaling, the node can allocate additional resources — obtaining, for example, virtual machines (VMs) as child nodes — and then it can deploy processing operations on these VMs.

Besides the elastic data processing capabilities, each single node also maintains data storage (such as a database) to realize stateful operations (such as data aggregations) as well as a messaging infrastructure to receive data from sensors or preceding nodes. For the outgoing communication, the node further maintains a message buffer to ensure that no information is lost, if communication to succeeding nodes isn't possible due to temporary technical failures or environmental aspects, such as communication dead spots. The node's last component is the monitor, which records changing data rates over time to allow predictive reasoning for the identification of future computational resource requirements.

Open Research Challenges

Even though elastic stream processing builds on established principles and concepts, four important research challenges (detailed in the following) must be addressed to achieve widespread adoption. These topics include methodologies and algorithms as well as the software tools necessary to realize elastic stream processing for distributed environments.

Cost-Efficient Resource Provisioning

Although there has been much research toward the optimal deployment of stream processing operations on fixed resources, there are hardly any advances in leveraging the resource elasticity provided by cloud-based resources. Because the system load varies over time due to varying data rates, a stream processing system may run into over- or underprovisioning

Mobility Domain Scenario

Our motivation for elastic stream processing originates from the omnipresent trend towards smart cities. Smart cities rely on numerous sensors that cover different aspects of the city's management and interaction with citizens, such as smart energy, smart homes, and smart transport networks.[1] These smart systems form a distributed environment. To demonstrate the challenges for stream processing, we consider an intelligent transportation system (ITS).[2] The ITS is responsible for monitoring and directing public transport as well as managing related infrastructure, such as information displays. The ITS depends on different data sources — for example, GPS sensors mounted on public transport vehicles, traffic information (such as induction loops measuring the number of cars driving by), or weather information.

Figure A illustrates the orchestration of the ITS's different data sources and stream processing operations. Four value-adding services are depicted that are relevant for system users. The first two services represent simple services that notify the driver of a public transport vehicle about the latest traffic information. Each service consists of two stateless processing operations (see Figure A). The first operation filters traffic sensor data and the second operation propagates the information to the drivers.

More complex services, such as the update service for displays, obtain information from all four data sources, an additional operation (which calculates public transport vehicles' movement speed based on GPS locations over time), as well as routing information. The routing information is obtained from a cloud-based data repository. Movement speed and sensor information are then forwarded to the prediction operation, which reasons on the data. This prediction operation is stateful, because it stores historic predictions over a longer time span to provide better prediction results.

Besides the operational components, the ITS also provides real-time routing information for public transport. Public transport users initiate routing queries and three stateless operations process each query to obtain the desired result for the user.

The individual sensors' data rates (the amount of information provided by the sensor at a specific point in time) changes throughout the course of the day, mainly because of the commuters' different driving directions. These volatile data rates challenge a stream processing system to cope with varying resource requirements to process streaming data.

The stream processing system further obtains the information from different sensors and data repositories (such as roadmaps), which are located in different geographical locations. Because the data originate from different infrastructures, the data might be inconsistent across various sources: A GPS device reports a bus driving on a specific street while an induction loop on that same street detects a traffic jam.

When processing streaming data from various sources, it's challenging to select the most accurate and reliable data source. In terms of cost and performance optimization, it's desirable to process the data geographically from the closest location. This reduces the effort required, such as data transfer costs that are required to transfer streaming data.

This scenario provides a brief glance at services and challenges from the mobility domain. Similar use cases also arise in other domains, such as building automation for smart cities, which involves monitoring countless sensors and processing their data to ensure a safe and pleasant environment. We further envision similar challenges in other areas such as e-healthcare, where systems must process large amounts of data in real time to gain insight into a patient's time-dependent information, such as heart rate or temperature.

References

1. L. Fei et al., "Web-Scale Service Delivery for Smart Cities," *IEEE Internet Computing*, vol. 17, no. 4, 2013, pp. 78–83.
2. A. Biem et al., "Real-Time Traffic Information Management Using Stream Computing," *IEEE Data Eng. Bullet*, vol. 33, no. 2, 2010, pp. 64–68.

scenarios, if the system only has a fixed amount of resources at its disposal. For instance, in off-peak times, a stream processing system could run into an overprovisioning scenario, because not all available computational resources are used. However, a stream processing system may also run into an underprovisioning scenario at peak times, when the stream-processing system can't process all incoming requests or streaming data, and therefore violates SLAs. The cloud computing paradigm[4] provides a promising solution by replacing fixed resources with elastic ones. A resource-elastic stream processing system leases and releases the required computational resources on demand to minimize the operational costs while guaranteeing a high QoS.

Although the introduction of elastic resources provides a solution to resource-related challenges, the ad hoc nature of such resource-allocation approaches introduces a dimension of complexity for deploying stream processing operations. Also, the leasing and releasing strategy must be optimized for real-world requirements, such as the billing time unit (the minimal leasing duration for cloud resources) and historical data rates. Such data rates allow the stream processing system to predict future resource requirements and optimize allocation of elastic resources just in time to avoid underprovisioning scenarios.

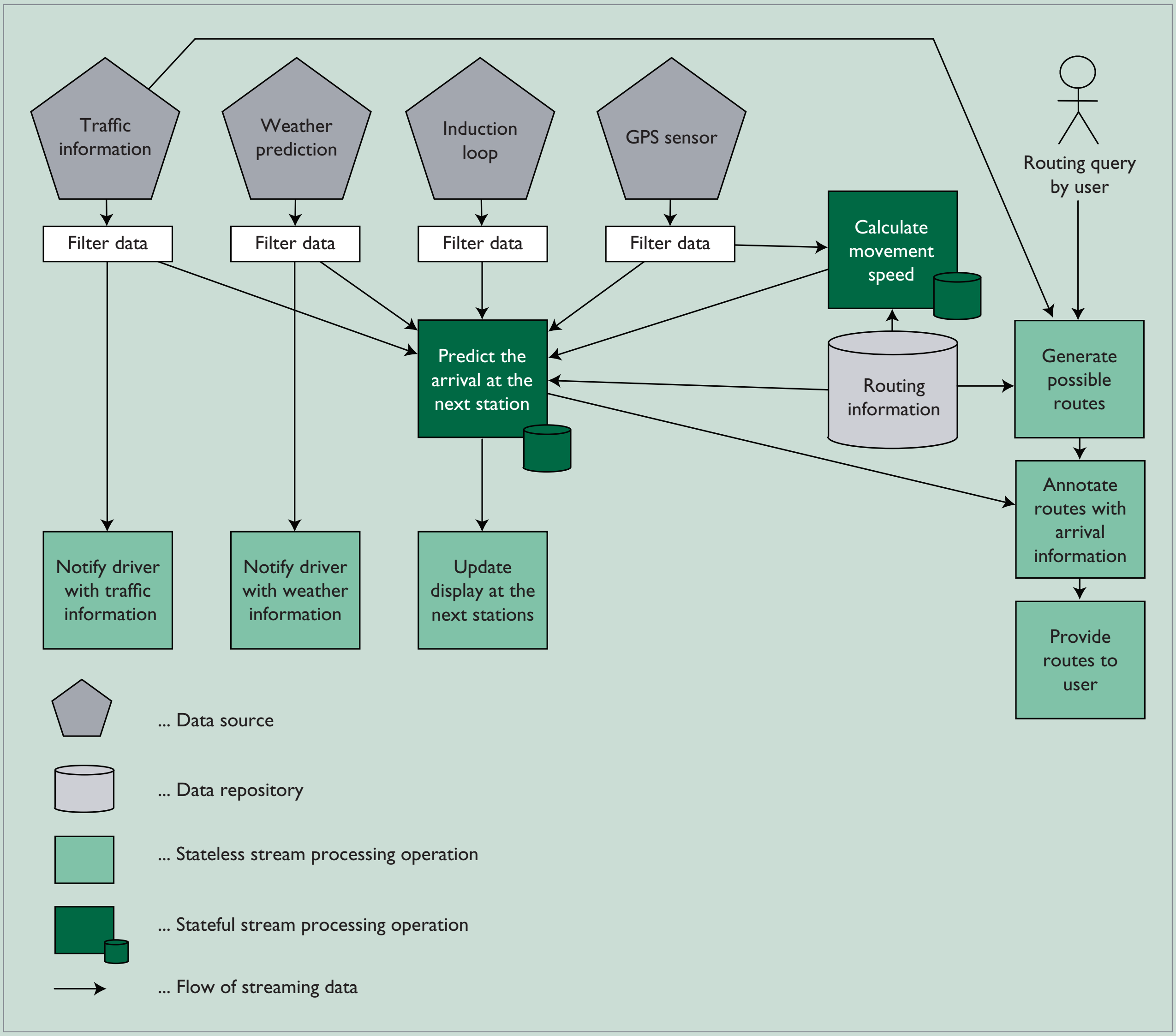


Figure A. Motivational scenario. Orchestrating the different data sources and stream processing operations provided by an intelligent transportation system (ITS).

SLAs' Data Stream Processing

Because quality elasticity represents one of the elasticity dimensions for stream processing, we must design appropriate SLAs to assess the QoS for elastic stream processing. Until now, stream processing only considered the *latency* of processed streaming data or the *ratio of dropped streaming data* in comparison to all streaming data.

Besides such basic SLAs, there are other areas that SLAs can cover: First, SLAs can be applied to queries on streaming data, making it possible to relax the result's quality in a controlled manner by shortening the observation time span or omitting relevant query data.

Second, SLAs can be assigned to individual streaming data, such as privacy restrictions or stating a validity period. In some domains, the information provided by sensors is transient and the information may be outdated after a few seconds. Therefore, it's possible to implement a context-sensitive load-shedding mechanism that discards all outdated data, instead of randomly discarding data to reduce the load.

Efficiently Placing and Migrating Stream Processing Nodes

Optimal placement of stream processing nodes is crucial for performance- and cost-efficient stream processing in the IoT. Different deployments are possible for stream processing nodes — for instance, on an embedded infrastructure with limited processing and

Related Work in Stream Processing

Because data rates change over time—in some cases even drastically—processing this data in real time isn't a straightforward task. To process streaming data, different stream processing systems such as Aurora,[1] Borealis,[2] MillWheel,[3] or IBM System S[4] provide solutions.

Early stream processing systems like Borealis[2] originate from the database-management domain. From a high-level perspective, such systems extend the relational database model to support the continuous aspects of streaming data. The Borealis architecture considers different processing nodes—such as software components that process streaming information—to enable parallel data processing for large amounts of data. However, the architecture only considers quality elasticity in terms of load shedding to deal with load peaks.

More advanced stream processing systems such as IBM System S[4] reliably and efficiently process different kinds of streaming data (such as financial or telecommunication data). Although the processing nodes support a distributed deployment (on a cluster, for example), IBM System S still maintains several centralized components, which manage monitoring, communication, and failure recovery. Besides research prototypes like Borealis, and proprietary stream processing systems such as IBM System S, there are also community-driven frameworks like Apache Storm (https://storm.apache.org) or Apache S4 (http://incubator.apache.org/s4/). Both frameworks can process huge amounts of data for Web-based services, to process social media data, for example. The architecture of these frameworks is based on a central component that manages data flow. Even though these stream processing frameworks feature a cluster-based deployment, they don't support resource elasticity for single processing nodes or an orchestration of processing nodes during runtime.

More recently, Benjamin Satzger and his colleagues[5] proposed an elastic stream computing platform for the Cloud, which provides an easy-to-use programming model to leverage the resource elasticity of the Cloud to deal with volatile data rates. However, their prototype doesn't address the challenges of accessing relevant data from different cloud infrastructures. Instead, they focus on elastically allocating computational resources in the Cloud for one static data stream processing application.

References

1. D. Carney et al., "Monitoring Streams: A New Class of Data Management Applications," *Proc. 28th Int'l Conf. Very Large Data Bases*, 2002, pp. 215–226.
2. D.J. Abadi et al., "The Design of the Borealis Stream Processing Engine," *Proc. Conf. Innovative Data Systems Research*, vol. 5, 2005, pp. 277–289.
3. T. Akidau et al., "MillWheel: Fault-Tolerant Stream Processing at Internet Scale," *Proc. Very Large Data Bases Endowment*, vol. 6, no. 11, 2013, pp. 1033–1044.
4. L. Amini et al., "SPC: A Distributed, Scalable Platform for Data Mining," *Proc. 4th Int'l Workshop on Data Mining Standards, Services, and Platforms*, 2006, pp. 27–37.
5. B. Satzger et al., "Esc: Towards an Elastic Stream Computing Platform for the Cloud," *IEEE Int'l Conf. Cloud Computing*, 2011, pp. 348–355.

storage capabilities next to the sensor; on clusters with fixed computational resources; or on a cloud infrastructure that provides unlimited computational resources and unlimited data storage.

Besides the volatile resource requirements, it's also important to consider the communication channels between the different nodes' locations, as well as relevant SLAs (for example, the maximum processing time for streaming data or the ratio of discarded streaming data). These different aspects form a nondeterministic polynomial time (NP)-complete problem for an optimal deployment solution in terms of costs, performance, and SLAs. Along with the optimal initial deployment, it's important to investigate robust migration strategies for processing nodes. Because the IoT landscape is constantly evolving, we might need to migrate processing nodes from one geographical location to another during runtime. This requires a sophisticated migration strategy to ensure a consistent state for stateful stream processing nodes while maintaining continuous uptime and responsiveness.

Data Elasticity

Optimal and efficient access to the data from remote data repositories is required, but this efficiency gain must not reduce the streaming data's quality and availability. Because the data originates from geographically distributed as well as inconsistent or even conflicting data sources, we must develop methodologies for stream processing to consolidate the streaming data at runtime. These methodologies should be robust when data are noisy, uncertain, delayed, or (at worst) unavailable. Current advances in Semantic Web technologies represent one possible solution approach that could help define requirements on data and its representation. To achieve this, the level of description must be minimal to avoid overloading the stream processing system, which aims to manipulate light objects with only minimal operational overhead.

Based on current state-of-the-art stream processing and the future challenges we anticipate with the IoT, we identified several open areas for research in elastic stream processing for distributed environments. Although there's room for improvement, elastic stream processing can handle the challenges issued by the IoT to realize smart systems.

Acknowledgments

This article is supported by Vienna University of Technology research funds. This work is partially supported by the Commission of the European Union within the SIMPLI-CITY FP7-ICT project (grant agreement 318201).

References

1. M. Cherniack et al., "Scalable Distributed Stream Processing," *Proc. Conf. Innovative Data Systems Research*, 2003, vol. 3, pp. 258–268.
2. S. Acharya et al., "The Aqua Approximate Query Answering System," *ACM Sigmod Record*, 1999, vol. 28, no. 2, pp. 574–576.
3. G. Copil et al., "SYBL: An Extensible Language for Controlling Elasticity in Cloud Applications," *Proc. 13th Int'l Symp. Cluster, Cloud, and Grid Computing*, 2013, pp. 112–119.
4. M. Armbrust et al., "Above the Clouds: A Berkeley View of Cloud Computing," *Comm. ACM,* 2010, vol. 53, no. 4, pp. 50–58.

Christoph Hochreiner is a PhD student at the Distributed System Group at the TU Wien, Austria. His research interests cover the whole spectrum of cloud computing, specifically stream processing systems, service monitoring, and security implications. Hochreiner has an MSc in software engineering and Internet computing from the Vienna University of Technology. Contact him at c.hochreiner@infosys.tuwien.ac.at; www.infosys.tuwien.ac.at/staff/hochreiner/.

Stefan Schulte is an assistant professor at the Distributed Systems Group at TU Wien and the project manager of the ongoing European Union's Seventh Framework Programme project SIMPLI-CITY – The Road User Information System of the Future (http://www.simpli-city.eu). His research interests span the areas of cloud computing and service-oriented computing, with a special focus on aspects of quality of service. Schulte has a PhD in Web Service discovery from the Technical University of Darmstadt. Contact him at s.schulte@infosys.tuwien.ac.at; www.infosys.tuwien.ac.at/staff/sschulte/.

Schahram Dustdar is a full professor of computer science and he heads the Distributed Systems Group at TU Wien. His work focuses on Internet technologies. Dustdar is a member of the Academy Europeana, an ACM Distinguished Scientist, and recipient of the IBM Faculty Award 2012. Contact him at dustdar@dsg.tuwien.ac.at; dsg.tuwien.ac.at/.

Freddy Lecue is a research scientist and lead investigator in large-scale reasoning at IBM Research–Ireland. His research focuses on AI, knowledge representation and reasoning, service computing with a focus on Semantic Web technologies, and (hybrid) reasoning from various domains. Lecue has a PhD in computer science from Ecole des Mines de Saint-Etienne, France. Contact him at freddy.lecue@ie.ibm.com; http://researcher.watson.ibm.com/researcher/view.php?person=ie-freddy.lecue.

This article originally appeared in IEEE Internet Computing, *vol. 19, no. 6, 2015.*

From the Editor in Chief

Editor in Chief: Maria R. Ebling ■ IBM T.J. Watson Research Center ■ ebling@us.ibm.com

Turning Back the Clock

Maria R. Ebling, *IBM T.J. Watson Research Center*

Where was *IEEE Pervasive Computing* 10 years ago? What were our special issue topics, and what were the top papers from each of those issues? What were the burning issues, and where do those issues stand today? Let's turn back the clock.

TEN YEARS AGO

In 2005, *Pervasive Computing* was only in its fourth year, a toddler. Mahadev Satyanarayanan, our founding father, was still at the helm.

Our January–March issue focused on "Energy Harvesting and Conservation." Our top-cited article from this issue was Joseph Paradiso and Thad Starner's "Energy Scavenging for Mobile and Wireless Electronics." Satyanarayanan's argument that "radical improvements in battery technology are unlikely"[1] certainly rings true. Battery life remains an important issue. In fact, at our editorial board meeting this past June, we selected energy harvesting as a theme for an upcoming special issue.

Our April–June issue focused on "The Smart Phone." Satyanarayanan observed that the most common pervasive device was the cell phone,[2] and the issue examined what would come next. Now, 10 years later, that has become clear. Smartphones are, indeed, ubiquitous. They include both PDA capabilities and email access, but so much more with music, games, health and fitness, and the ability to manage membership cards. The only thing my smartphone does not yet manage is my keys, and I think that is only a matter of time. Our top-cited article from this issue was "ContextPhone: A Prototyping Platform for Context-Aware Mobile Applications" by Mika Raento, Antti Oulasvirta, Renaud Petit, and Hannu Toivonen. I would argue that this is pretty much a solved problem at this point, with mobile apps having access to a plethora of context information.

With the Deflategate scandal from the AFC Championship Game earlier this year, it is clear that sensing in sports is not yet a solved problem.

Our July–September 2005 issue focused on "Sports Technologies." This issue examined the use of pervasive sensing in sports, from equipment, to athletes, to sports governance. With the Deflategate scandal from the American Football Conference (AFC) Championship Game earlier this year,[3] it is clear that sensing in sports is not yet a solved problem. In fact, as with energy harvesting, our editorial board meeting identified sports and fitness as a theme topic for a future issue.

Our October–December 2005 issue focused on "Rapid Prototyping." Our most-cited article was written by Feng Zhu, Matt Mutka, and Lionel Ni on "Service Discovery in Pervasive Computing Environments." Looking at the literature, service discovery certainly remains a common topic for papers in mobile and pervasive computing, although service discovery has also made significant progress, such as the ease with which we can now connect headsets via Bluetooth or phones and laptops to Wi-Fi networks. Rapid prototyping, though not a solved problem, has also become easier. Just look at the progress people make during very short-term hackathons!

I hope you enjoyed this trip down memory lane as much as I did. I found it fascinating to see what we were doing 10 years ago, realizing just how far we have come! This year's special issues have focused on "Privacy and Security," "Smart Spaces," "Digitally Enhanced Reality," and "Pervasive Food." Some of these topics could easily have been topics of interest a decade ago, and others I think are well beyond where we were back then, especially this issue's focus on food.

IN THIS ISSUE

In addition to our theme articles on pervasive food, we also have two feature articles.

MISSION STATEMENT: ***IEEE Pervasive Computing*** is a catalyst for advancing research and practice in mobile and ubiquitous computing. It is the premier publishing forum for peer-reviewed articles, industry news, surveys, and tutorials for a broad, multidisciplinary community.

In "Classifying Text-Based Computer Interactions for Health Monitoring," Lisa Vizer and Andrew Sears examine the differences in typing behaviors by older adults with and without early cognitive impairment. They look at typing speeds and pauses between keystrokes, as well as the complexity of the language used and the number of references to other people. Their results show a promising mechanism to identify (and potentially measure the progress of) cognitive impairment in older adults using a common, everyday task. If this mechanism bears fruit, we might have a way to routinely monitor cognitive impairment in a minimally invasive manner.

For our second feature article, Mitja Luštrek, Hristijan Gjoreski, Narciso González Vega, Simon Kozina, Božidara Cvetković, Violeta Mirchevska, and Matjaž Gams present "Fall Detection Using Location Sensors and Accelerometers." In this article, the authors describe a system that uses wearable location sensors, enhanced with accelerometer sensors, to detect falls with surprising accuracy. Given the demographics of our aging population around the developed world, solutions such as this will prove important as we learn to help our elders (including, in many cases, ourselves) live independently longer and with fewer care providers.

We also have several interesting departments this issue. In our Smartphones department, Lori Flynn and Will Klieber from CERT discuss security issues as they relate to smartphones. They point out the amount of sensitive data, both personal and business, stored on smartphones and the need for vigilance. They examine all of the common smartphone platforms and operating systems. I strongly encourage you to take the time to read through this.

In our Conferences department, Ana Nika, Sofia Scataglini, and Sirine Taleb provide a fantastic overview of MobiSys 2015. Some interesting highlights include a keynote by Krishan Sabnani (Research VP at Bell Labs), a ring that detects typing on a non-existent keyboard from Shahriar Nirjon (HP Labs), and a joint session between WWW and MobiSys.

In our Innovations in Ubicomp Products Department, Lars Lischke, Dominik Weber, and Scott Greenwald observe that we now also have "boards" in widespread use and not just the "tabs" and "pads" from Weiser's vision.[4] As someone who does not pay much attention to the latest television technology, such as "Apple TV" and "Chromecast," I enjoyed learning about the smart TV "boards" and their interactions with the "tabs" and "pads" of today. I also found the discussion of "Android Auto" an interesting foreshadowing for our upcoming special issue on smart vehicles.

Our Notes from the Community department looks at a wide variety of topics. I find the use of location technology by the NFL interesting, but I would have expected to find the technology in the footballs rather than on the players after last year's "deflategate" scandal. I have to admit that I have a very negative reaction to the Jewelbots. Though middle-school girls might think they are very cool (and the Jewelbots might sell like hotcakes), the new opportunities for bullying are very real. I also found the discussion about Aibo being on the endangered species list to open an important topic of conversation. As we look at robots to be companions for our aging population, we really need to consider the ability of those seniors to adjust to new "companions." Are there ways we could make the "personality" portable so that it can be moved into a new robot as the technology improves so that our senior citizens do not have to grieve the loss of another "friend"? I have mixed feelings about the "Do Not Fly Over" list. I appreciate the attempt to "do the right thing," but I fear it will work about as well as the "Do Not Call" list does with chimney sweep companies! These are but a few of the topics covered in this issue's column.

Finally, in our Pervasive Health department, Michael Gonzales and Hanna Schneider provide a thorough overview of the Pervasive Health conference, which was held this past May in Turkey. From the keynotes to the technical sessions, it sounds like a wide variety of healthcare-related work was presented. I look forward to the day when some of these technologies prove successful enough to help patients in the real world.

Reviewing how far we've come makes me wonder where we'll be in another 10 years. Will the vision of robotic companions help address the challenges surrounding an aging

PERVASIVE BORED AND BRILLIANT CHALLENGE

This issue's challenge is to consider how an electronic reading device, such as a Kindle, enhances or detracts from boredom. Is such a device simply a lighter replacement for a paperback book that you would otherwise carry around? Or is it something that you find yourself using for time-wasting activities?

These are questions to which I have been giving much thought since I initially heard the NPR "Bored and Brilliant" series (www.wnyc.org/series/bored-and-brilliant). I own a smartphone, but I also own an electronic book. I do not read books on my smartphone because I prefer the experience of my eBook. Based on my observations of my own eBook-reading behavior, I use my eBook at times and places where I would have picked up a paper-based book. I don't reach for my eBook every spare moment in the same way that I do my phone. In fact, I often do not carry my eBook with me. I much prefer my eBook to paper-based books simply because of the weight of some of the books I like to read. I also prefer it when I travel because I can bring a number of titles with me for the same investment in space and weight. I must admit that in considering these questions for myself, I do not consider my eBook in the same class as my smartphone in terms of avoiding boredom.

Returning back to last issue's challenge of deleting time-wasting apps from your phone, I have identified the next game that needs to go … now I just have to hit the delete key!

This article originally appeared in
IEEE Pervasive Computing, *vol. 14, no. 4, 2015.*

population? Will autonomous cars be commonplace? Where will we be on energy generation and consumption? What device might displace the smartphone? My personal opinion is that we will make some pretty impressive progress on home robots and autonomous vehicles, in part because the demographics demand that we do. I fear that the battery issues will still be challenging us. For my generation, I expect a smartwatch could displace the smartphone or perhaps a smart "pin" (similar to Dick Tracy's watch or the Star Trek tricorder). Only time will tell.... ■

Maria R. Ebling is a director at the IBM T.J. Watson Research Center. She manages a team building systems capable of supporting a Smarter Planet while not forgetting about the people who use such systems. Ebling received her PhD in computer science from Carnegie Mellon University. She's a member of the IBM Academy of Technology, a distinguished member of the ACM, and a senior member of IEEE. Contact her at ebling@us.ibm.com.

REFERENCES

1. M. Satyanarayanan, "Avoiding Dead Batteries," *IEEE Pervasive Computing*, vol. 4, no. 1, 2005, pp. 2–3; http://doi.ieeecomputersociety.org/10.1109/MPRV.2005.5.
2. M. Satyanarayanan, "Swiss Army Knife or Wallet?" *IEEE Pervasive Computing*, vol. 4, no. 2, 2005, pp. 2–3; http://doi.ieeecomputersociety.org/10.1109/MPRV.2005.39.
3. "DeflateGate: A Timeline of the Patriots Scandal Involving Star QB Tom Brady," *Daily News*, 29 July 2015; www.nydailynews.com/sports/football/deflategate-timeline-latest-patriots-scandal-article-1.2213285.
4. M. Weiser, "The Computer for the 21st Century," *Scientific American*, vol. 265, no. 3, 1991, pp. 66–77.

Ask Your Doctor … About Computers

Joseph November
University of South Carolina

Editor: Bradley Fidler

Do you ever get the feeling that your healthcare providers are decades behind the times when it comes to information technology? Does it trouble you that your credit score is instantly accessible around the country but your health record cannot even follow you around the block? If the answer to these questions is "yes," then the time may be right to ask your doctor about computers. How might a chat with your doctor about computers go? What you hear will likely be mostly negative. "Doctors say electronic records waste time," ran one *US News & World Report* headline, which was followed by the claim that "technology has slowed doctors' work."[1]

As historians of computing, we have the resources not only to help the general public think more constructively about the roles of computers in medicine, but also to see how what is happening in medicine lays bare the great and small changes the use of information technology has brought to everyday life. More specifically, we can answer an important question that lies, in part, behind primary care physicians' dislike of electronic records: How did we wind up with computers in most doctors' offices but little communication between these computers?

Toward Electronic Health Records

Your physician's chief complaint will likely include mention of the HITECH (Health Information Technology for Economic and Clinical Health) Act and of Epic Systems. The HITECH Act, a major part of the American Recovery and Reinvestment Act of 2009 (aka The Recovery Act), mandated that the US Department of Health and Human Services (HHS) spend on the order of $25 to $45 billion to create a national, electronic network for medical records. American healthcare providers, ranging from large hospital systems to small private practices, were granted these HHS funds to start using computers to create an electronic health record (EHR) for each patient. For proponents of President Obama's plan to provide health coverage for all Americans, an effective EHR was hailed as a means to cut costs while at the same time improving the quality of care. With the HITECH Act in place, Americans had, in principle, a mechanism to encourage (via generous grants) healthcare providers to go digital and punish (via steep fines) those who would not. Proving compliance, particularly to electronic security and privacy measures mandated by the act, can be a nightmarish process.

The most widespread EHR software the HITECH Act encouraged—or forced—physicians to adopt are products of the privately held Epic Systems Corporation, which developed software that manages the health records of most Americans (approximately 170 million people). For the purpose of bringing a given institution into compliance with the HITECH Act, Epic's products have a reputation for getting the job done. Nevertheless, they are also notoriously expensive and ineffective when it comes to sending heath records from one institution to another, even when both are using Epic software.[2] Epic's interoperability problems, coupled with the theme-park atmosphere at Epic's "Intergalactic Headquarters," generated fierce criticism of the company across the political spectrum and widespread mockery of its slogan "Do good. Have fun. Make money."[3]

By some measures, the carrot-and-stick approach of the HITECH Act worked. In 2008, less than one in five doctors in the US kept some form of electronic records. By late 2015 that proportion had risen to more than four in five.[4] Hospitals have seen an even more dramatic rate of EHR adoption. In 2009, just 9 percent of hospitals used EHRs, whereas today that figure exceeds 75 percent.[5] The clinic may have become home to a "strange tribe" of compulsive computer users who manufacture database records, but there is an odd limit to this activity.[6] Health records, even if they are kept locally in entirely digital format, still are generally faxed or mailed between clinics. Furthermore, American patients simply do not have a centralized health record any care provider could access.

MUMPS in Your Medical History

The origins of this situation is not just or even mainly about protecting patients' privacy—by 2016 most American adults have become accustomed to handling private personal information online. Instead, the answer lies in the history of computing. Epic and many of its counterparts are built around the Massachusetts General Hospital Utility Multi-Programming System (MUMPS), which was developed in the late 1960s. MUMPS is not so much a general-purpose programming language as it is a hierarchical database construction kit with a built-in programming language. For readers familiar with database architecture, MUMPS was NoSQL decades before the concept became trendy.[7] Although MUMPS was much easier to learn than, say, Fortran or Cobol, it is a clunky tool. The kludges one can put together with MUMPS can provide highly functional solutions to particular data-management problems, but they cannot practically be ported as solutions to related problems and they generally do not interface well with other systems, even other MUMPS-based systems.[8]

The story of the development of MUMPS during the late 1960s is one of accommodation and displaced triumph. For the doctors at Mass General, where it was created, using MUMPS to create an EHR system tailored to their specific decision-making behavior, data needs, and institutional rules let them keep an electronic health record while making little compromise in terms of their authority or the way they worked. For the National Institutes of Health, which had funded Mass General's early foray into computerized health records, MUMPS was seen as a tentative first step toward a grand vision of computerized, rationalized, centralized medicine.[9] Starting in the 1960s, the NIH had hoped that by introducing computers to medical research and clinical care they could compel physicians to standardize their approaches to diagnosis and treatment.

During the 1970s and 1980s, MUMPS' versatility as a tool to build custom-made databases was seized upon by the few hospital systems that computerized; they sought the speed and convenience of electronically keeping dynamic records but also sought to avoid the imposition of external standards on their operations. Thus, a MUMPS-based system built for one hospital system could not be useful for another clinic, which operated under different rules. Companies like Epic Systems got good at building such customized systems, continuing that activity to the present day. Even the company's newest systems are built around InterSystems' proprietary upgrade to MUMPS, Caché.[10] Until recently, though, the EHR interoperability issue was thoroughly masked by the larger issue of the vast majority of care providers simply not using computers at all to create, store, and share records.

It may be tempting reduce America's EHR problem to an infection of MUMPS. It is certainly easy to point out the flaws of MUMPS as an idiosyncratic programming language and then decry the anticompetitive behavior of entrenched MUMPS-peddlers like Epic and their shortsighted enablers in the federal government. But the heavy use of MUMPS has also been found in—and arguably helped to nurture—environments in which interoperability has come to be taken for granted. In finance, for instance, where regulatory mechanisms force much standardization, MUMPS flourished, though its name was changed to "M" for marketing and morale reasons. By the late 1990s, M enjoyed a reputation as reliable tool for the ACID (Atomic, Consistent, Isolated, and Durable) transaction processing that make a big part of the day-to-day life of banking and stock trading.

In Need of Standardization

So if MUMPS is not toxic per se, then what is the cause of America's EHR illness? For Dr. Lawrence Weed, the answer is simple and ties back to his decades-long struggle to get physicians to think systematically about medical record keeping since the 1960s, when the *New England Journal of Medicine* published his groundbreaking study, "Medical Records that Guide and Teach."[11] After 40 years, Weed's study remains widely read but largely unheeded. The problem with health records, Weed discovered in the process of creating his own EHR system, PROMIS, was that it is impossible to make them consistent or even useful to doctors outside of one's own clinic if there is no universally agreed-upon standard of care.[12] Without this standard, several basic acts are impossible across clinical boundaries: deciding what goes into the record, determining which data are important, agreeing on whose opinion carries more weight, and building a mechanism to allow a record to be changed.

In the absence of a standard of care, Weed argued that an irrational "non-system" has emerged in American medicine.[12] Imposing a vast computer network onto this non-system has not, so far, brought the rise in quality or the drop in the costs of medicine that many Americans had hoped for. Moreover, the near omnipresence of the computer has not realized the now-60-year-old dream of bringing strict rationality to medical decision making through physicians' use of computers. If anything, making complex computer programs part of everyday medical life has brought even more absurdity to the non-system, which now has to accommodate the great expense of EHR software. Steven Teles' "kludgeocracy," to which Evan Hepler-Smith introduced *Annals* readers, has crossed from metaphor into reality and is thriving there.[13]

Given the deep divisions over the Obama Administration's attempts to implement a single-payer healthcare system in the US, it is unlikely that Americans are going to collectively agree on a standard of care anytime soon. Weed's non-system is likely here to stay. Whether information technology makes living with it more or less difficult will be a function of patients' abilities to encourage healthcare providers to make responsible choices about working with computers.

Being sufficiently familiar with the history of computing to see past the rhetoric and hype surrounding medical computing would go a long way toward getting Americans to steer medicine in a more efficient, more humane direction.

References and Notes

1. K. Leonard, "Doctors Say Electronic Records Waste Time," *US News & World Report*, 8 Sept. 2014.
2. S. Garber et al., "Redirecting Innovation in U.S. Health Care: Options to Decrease Spending and Increase Value," research report 308, Rand Corp., 2014; www.rand.org/content/dam/rand/pubs/research_reports/RR300/RR308/RAND_RR308.casestudies.pdf.
3. P. Caldwell, "Epic Fail: Digitizing America's Medical Records Was Supposed to Help Patients and Save Money. Why Hasn't That Happened?" *Mother Jones*, Nov./Dec. 2015; www.motherjones.com/politics/2015/10/epic-systems-judith-faulkner-hitech-ehr-interoperability.
4. B. Monegain, "More Than 80 Percent of Docs Use EHRs," *Healthcare IT News*, 18 Sept. 2015; www.healthcareitnews.com/news/more-80-percent-docs-use-ehrs.
5. Office of the Nat'l Coordinator for Health Information Technology, "Adoption of Electronic Health Record Systems among U.S. Non-Federal Acute Care Hospitals: 2008–2014," ONC data brief 23, Apr. 2015; https://www.healthit.gov/sites/default/files/data-brief/2014HospitalAdoptionDataBrief.pdf.
6. N. Ensmenger, "The Digital Construction of Technology: Rethinking the History of Computers in Society," *Technology and Culture*, vol. 53, no 4, 2012, pp. 753–776.
7. C. Mohan, "History Repeats Itself: Sensible and NonsenSQL Aspects of the NoSQL Hoopla," *Proc. 16th Int'l Conf. Extending Database Technology*, 2013, pp. 11–16; doi:10.1145/2452376.2452378.
8. As coding humorist Alex Papadimoulis tartly put it, MUMPS "is conceptually different from just about every other programming language out there. The primary design goal MUMPS was to create code that, once written, can never be maintained by anyone ever again." A. Papadimoulis, "A Case of the MUMPS," *Daily WTF*, 13 Feb. 2007; http://thedailywtf.com/articles/A_Case_of_the_MUMPS.
9. R.S. Ledley and L.B. Lusted, "Reasoning Foundations of Medical Diagnosis," *Science*, vol. 130, no. 3366, 3 July 1959, pp. 9–21.
10. Caché also provides the framework for the Veterans Administration's Veterans Health Information Systems and Technology Architecture (VistA)
11. L. Weed, "Medical Records that Guide and Teach," *New England J. Medicine*, vol. 278, 14 Mar. 1968, pp. 593–600.
12. L. Weed and L. Weed, *Medicine in Denial*, CreateSpace Independent Publishing Platform, 2011, p. 1.
13. E. Hepler-Smith, "Simple Problems," *IEEE Annals of the History of Computing*, vol. 36, no. 1, 2014, pp. 88, 85–87.

Joseph November *is an associate professor and McCausland Fellow in the Department of History at the University of South Carolina. Contact him at november@sc.edu.*

This article originally appeared in
IEEE Annals of the History of Computing,
vol. 38 no. 1, 2016.

EDITOR: Irena Bojanova, US National Institute of Standards and Technology, irena.bojanova@computer.org

© Tadeusz Ibrom | Dreamstime.com

IT Project Failures

What Management Can Learn

Jay Liebowitz, *Harrisburg University of Science and Technology*

IT project failures are omnipresent. Let's look at just a few global examples[1,2]:

- The UK's Home Office sank nearly 350 million pounds into a computer system that had to be scrapped in 2013.
- In 2014, the UK government reported that successful delivery of 41 of the 199 projects in its 500 billion pound portfolio was "in doubt."
- The annual cost of failed US government IT projects is estimated to be as high as $20 billion (out of the $80 billion the US federal government spends annually on IT projects).
- The well-known HealthCare.gov Web portal was an IT disaster, with a "trauma team" needed to salvage the work.
- The US Air Force's Expeditionary Combat Support System for consolidating more than 200 legacy systems spent $1.1 billion taxpayer dollars before being terminated.
- The National Program for IT in the UK failed terribly after 12.7 billion pounds was spent to create an online portal to allow citizens to access their personal health information (the project was four years late).
- A smartcard ticketing system for public transportation in Australia was implemented years late and went US$500 million over budget.

The list goes on and on—not to mention IT failures that recently took down United Airlines and the *Wall Street Journal* website, and halted the New York Stock Exchange for four hours.[3] We might as well throw in the two US Office of Personnel Management (OPM) IT debacles in which 4.2 million individuals had their personnel data stolen in April 2015; then in July 2015, 21.5 million individuals had sensitive information stolen from the background investigation databases.[4] What a mess!

Categorizing IT Project Failures

Most project managers, IT or otherwise, will claim that the "iron triangle" of scope, cost, and time seems to be the Holy Grail of projects (see Figure 1).

IT projects often encounter these dilemmas, but we can categorize failures by looking at process-driven, context-driven, and content-driven issues.[5]

Process-driven failures are those relating to business planning, project planning, project management and control, strategic formulation, and the change-management process. Content-driven failures relate to the information system itself as a technology, system design, and related technology issues. Context-driven failures relate to the environment in which the project is being developed, such as the firm's culture, structure, top management style, internal communications, user involvement, absence of an agent of change, politics, and reactive versus proactive style for handling problems.

From Rateb Sweis's research,[5] we learn that IT projects may fail most, in priority order, due to a high degree of customization, changes late in the design stage, and underestimating the timeline. This seems to be consistent with most of the other IT project failure research in the field.

Here, I focus on context-driven issues overlapping with some of the process-driven ones.

Let's see what context-driven issues result from a management

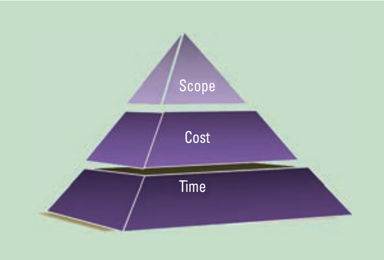


Figure 1. The "iron triangle" of IT project failures. IT project managers must balance the scope, cost, and time elements of an IT project very carefully.

perspective. One issue is organizations tending to start by picking the solution they like the look of, rather than fully evaluating the business case.[6] Even though a thorough business case is conducted, there might sometimes be a business case that does not follow best practice and can be severely affected by managers wanting to influence the process of project approval (that is, "I want this project to happen" could outweigh the poor business case).[6]

Another related context-driven issue deals with IT project complexity, and we are often very poor at handling complex tasks. Coupled with complexity could be the feeling that IT lacks the accepted ownership of the project.

Moreover, lessons might not be learned from other IT project failures. Perhaps applying some of the knowledge management tenets here for learning from others could be useful. A key context-driven issue also deals with producing a product but with no business outcomes evident. In this sense, IT might be delivering something that the business really doesn't want or use.[6]

What Can Management Do?

One suggested solution to these issues is to better educate current and future IT project leaders. In fact, the UK is already doing this by requiring its leaders of major projects to be further educated on how to better manage them. The Major Projects Leadership Academy was established in 2012 through a partnership with Oxford University and Deloitte (www.sbs.ox.ac.uk/programmes/execed/custom/our-clients/major-projects-leadership-academy). Already, more than 120 project leaders have graduated from the academy (with a further 200 enrolled).[1]

Another idea is to use a more structured approach to decide whether to sign off on a new IT program. The Danish government has deployed a business case modeling package to help decision makers better evaluate the decision-making process.[6] Related to this, management should also perform reviews or project health checks in a structured way during all phases of the IT project.[6]

As previously hinted, IT project managers and their staff should apply knowledge management principles to capture, share, and apply their knowledge from previous IT projects to new ones. NASA has the Academy for Sharing Knowledge and Lessons Learned Information System/NASA Engineering Network, through APPEL (Academy for Program/Project Engineering Leadership), which allows NASA project managers to share lessons learned in project management, systems engineering, risk management, safety, and other relevant areas (http://appel.nasa.gov).

Finally, better coordination between business users, IT, and the finance department can help ensure that the IT project will be designed, developed, and implemented within scope, time, and cost. Without this close collaboration, the "field of dreams" may become just an empty field!

References

1. S. Hendershot, "Facing Up to Government IT Project Failures," Project Management Inst., May 2015; www.pmi.org/Learning/PM-Network/2015/facing-government-it-project-failures.aspx.
2. K. DeSouza and K. Smith, "Why Large Scale Government IT Projects Fail and What to Do about It," *The Health Care Blog*, 2 June 2015; http://thehealthcareblog.com/blog/2015/06/02/why-large-scale-government-it-projects-fail-and-what-to-do-about-it/.
3. R. Marsh, "Tech Fail! Explaining Today's 3 Big Computer Errors," *CNN Money*, 8 July 2015; http://money.cnn.com/2015/07/08/technology/united-nyse-wsj-down/.
4. "OPM Announces Steps to Protect Federal Workers and Others from Cyber Threats," news release, OPM Office of Communications, 9 July 2015; https://www.opm.gov/news/releases/2015/07/opm-announces-steps-to-protect-federal-workers-and-others-from-cyber-threats/.
5. R. Sweis, "An Investigation of Failure in Information Systems Projects: The Case of Jordan," *J. Management Research*, vol. 7, no. 1, 2015; www.macrothink.org/journal/index.php/jmr/article/viewFile/7002/5856.
6. N. Martindale, "Why Do IT Projects Fail?" *Economia*, 5 Jan. 2015; http://economia.icaew.com/business/january-2015/it-failures.

***Jay Liebowitz** is the Distinguished Chair of Applied Business and Finance at Harrisburg University of Science and Technology. He previously served as the Orkand Endowed Chair in Management and Technology at the University of Maryland University College and as a full professor at Johns Hopkins University. Contact him at jliebowitz@harrisburgu.edu.*

This article originally appeared in IT Professional, *vol. 17, no. 6, 2015.*

IEEE Computer Society 2016 Call for

MAJOR AWARD NOMINATIONS

Help Recognize Computing's Most Prestigious

IEEE Computer Society awards recognize outstanding achievements and highlight significant contributors in the teaching and R&D computing communities. All members of the profession are invited to nominate individuals who they consider most eligible to receive international recognition of an appropriate society award.

Computer Entrepreneur Award
Sterling Silver Goblet
Vision and leadership resulting in the growth of some segment of the computer industry.

Technical Achievement Award
Certificate/$2,000
Contributions to computer science or computer technology.

Harry H. Goode Memorial Award
Bronze Medal/$2,000
Information sciences, including seminal ideas, algorithms, computing directions, and concepts.

Hans Karlsson Award
Plaque/$2,000
Team leadership and achievement through collaboration in computing standards.

Richard E. Merwin Distinguished Service Award
Bronze Medal/$5,000
Outstanding volunteer service to the profession at large, including service to the IEEE Computer Society.

Harlan D. Mills Award
Plaque/$3,000
Contributions to the practice of software engineering through the application of sound theory.

Computer Pioneer Award
Silver Medal
Pioneering concepts and development of the computer field.

W. Wallace McDowell Award
Certificate/$2,000
Recent theoretical, design, educational, practical, or other tangible innovative contributions.

Taylor L. Booth Award
Bronze Medal/$5,000
Contributions to computer science and engineering education.

Computer Science & Engineering Undergraduate Teaching Award
Plaque/$2,000
Recognizes outstanding contributions to undergraduate education.

IEEE-CS/Software Engineering Institute Watts S. Humphrey Software Process Achievement Award
(Joint award by CS/SEI)
Plaque/$1,500
Software professionals or teams responsible for an improvement to their organization's ability to create and evolve software-dependent systems.

Deadline: 15 October 2016
Nomination Site: awards.computer.org
For more information visit: www.computer.org/awards

IEEE

IEEE computer society

Finding the Internet of Things–Related Job You Want

For this *ComputingEdge* issue, we asked Jaroslaw Domaszewicz, an assistant professor in Warsaw University of Technology's (WUT's) Faculty of Electronics and Information Technology and its Institute of Telecommunications, about career opportunities related to the Internet of Things (IoT). Domaszewicz's research interests include the IoT, pervasive computing, middleware, and the interaction between people and smart objects, and he has led WUT's team in a number of European Union projects in these areas. He is knowledgeable about pervasive-computing careers, especially those involving the IoT.

ComputingEdge: What careers in pervasive computing, as it relates to the IoT, will see the most growth during the next several years, and why?

Domaszewicz: In my opinion, the greatest career growth in pervasive computing and the IoT will occur in two areas: semantics and user interaction.

Interoperability between independently developed and deployed things and applications is a key prerequisite to realizing the IoT's vision. Currently, many sense-and-react applications are hardwired by the programmer to use specific sensing and actuating resources. The IoT's vision, on the other hand, is that applications should be able to discover new things, sensors, actuators, and data at runtime, and take advantage of them afterward. Semantic models and machine-interpretable semantic descriptions are what make such interoperability possible. Currently, there are a number of initiatives to develop semantic models for the IoT. I believe there will be a need for professionals who are knowledgeable in both fundamental semantic technologies and the dynamic landscape of IoT resource-description standardization efforts.

Meanwhile, as the IoT pervades human environments, users will be able to interact with more and more services. Today's typical user already engages with many smartphone and tablet applications. There are concerns, though, that the growing amount of human–machine interaction might distract users and harm their productivity and well-being. Thus, IoT-enabled services and smart objects must be designed so that user interaction is intuitive and nonintrusive. Most likely, there will be a growing need for professionals who are knowledgeable in the related fields of user-centered design, user-experience design, and usability evaluation.

ComputingEdge's Lori Cameron interviewed Domaszewicz for this article. Contact her at l.cameron@computer.org if you would like to contribute to a future *ComputingEdge* article on computing careers. Contact Domaszewicz at domaszew@tele.pw.edu.pl.

APPLICATION DEVELOPER: PETERSON TECHNOLOGY PARTNERS INC. seeks qualified application developer for its headquarters located in Rolling Meadows, IL & various & unanticipated work locations throughout the U.S. Resp. for designing & implementing mobile-based services & web-based solutions to provide optimum interface w/ mobile/eCommerce website for clients. Master's degree IT Engg, Info System Techn, IT Mgmt, Industrial Tech, Bus Admin or a closely related field of study (Will accept Bachelor's degree in the above fields plus 5 yrs related progressive exp in lieu of Master's degree) ea. alternative degree requirement w/ an academic or industrial background in (i) database, data mining, data warehouse, telecommunications (wireless, mobile & internet comm protocol), computer architecture & internet security (policies & risk mgmt strategies); & (ii) developing technical mobile/eCommerce solutions using HTML5, CSS3, Android, iOS, Angular Js, REST API, & coordinate Quality Assurance efforts. An EOE. Respond by mail to Peterson Technology Partners, 1600 Golf Rd, Ste 1206, Rolling Meadows, IL 60008. Refer to ad code: PTP-0216

SR. ETL DEVELOPER: designs & devel. data models; implem. ETL solutions; design, code & enhance software. BS in CS, EE or related, + 5 yrs of relev. IT exp. Email sarah.ulep@laureate.net w/ Job# 12450BR in subj. line. Laureate Education, Inc. 650 S. Exeter St., Baltimore MD 21202. EOE

SR. SOLUTN INTGRTN ARCHTCT (Plano, TX) Provide front end support on Clarity PPM, CA Business Intelligence, advanced reporting (Jaspersoft) installation & upgrade for Clarity PPM & PMO Accelerator. Address complex questions on functionality and product usage. Use DB SQL, DB Oracle, Java, GEL & scripting to provide workaround solutions. Liaise w/ the sftwr dvlpmnt team to report faults. Investigate tech problems and provide tech leadership. REQS: 5 yrs exp in job and/or a rel occup. Must have exp w/ Clarity PPM (functional & technical), CA Business Intelligence, Advanced Reporting (Jaspersoft), Installation & Upgrade for Clarity PPM, PMO Accelerator; DB SQL, DB Oracle, Java, GEL; Providing technical leadership across the Clarity PPM product line, fostering the tech community, identifying cross training opportunities & driving technical reviews of knowledge mngmnt content; Reading, understanding,& analyzing Clarity PPM Product Java source code; Performing Problem Source Identification (PSI) through use of code debuggers & inspection of code logic; Using scripting languages to provide

Cisco Systems, Inc. is accepting resumes for the following positions:

CARLSBAD, CA: Software Engineer (Ref.# CARL1): Responsible for the definition, design, development, test, debugging, release, enhancement or maintenance of networking software. **Software/QA Engineer (Ref.# CARL4):** Debug software products through the use of systematic tests to develop, apply, and maintain quality standards for company products.

CENTENNIAL, CO: Network Consulting Engineer (Ref.# CENT23): Responsible for the support and delivery of Advanced Services to company's major accounts. Travel may be required to various unanticipated locations throughout the United States.

HOUSTON, TX: Systems Engineer (Ref.# HOU1): Provide business-level guidance to the account team or operation on technology trends and competitive threats, both at a technical and business level.

LA PALMA, CA: Software Engineer (Ref.# PALM1): Responsible for the definition, design, development, test, debugging, release, enhancement or maintenance of networking software. Telecommuting permitted and Travel may be required to various unanticipated locations throughout the United States.

LINDON, UT: Software Engineer (Ref.# LIN1): Responsible for the definition, design, development, test, debugging, release, enhancement or maintenance of networking software.

MIAMI, FL: Solutions Integration Architect (Ref.# MIA107): Responsible for the development of the overall vision that underlies the customer-centric solution. Telecommuting permitted.

MOORESTOWN, NJ: Solutions Integration Architect (Ref.# MOO4): Perform analysis and diagnosis of highly complex networking problems and network designs. Travel may be required to various unanticipated locations throughout the United States.

RESEARCH TRIANGLE PARK, NC: Customer Support Engineer (Ref.# RTP1): Responsible for providing technical support regarding the company's proprietary systems and software. **Manager, Technical Support (Ref.# RTP25):** Responsible for leading a team in the delivery of world-class customer support on a line of products or for a targeted group of customers. **Consulting Systems Engineer (Ref.# RTP30):** Provide specific end-to-end solutions and architecture consulting, technical and sales support for major account opportunities at the theater, area, or operation level.

SAN FRANCISCO, CA: Software Engineer (Ref.# SJ3): Responsible for the definition, design, development, test, debugging, release, enhancement or maintenance of networking software. **CNG Staff (Ref.# SJ13):** Provide quality technical support for our client and partner base. **Software Engineer (Ref.# SF88):** Responsible for the definition, design, development, test, debugging, release, enhancement or maintenance of networking software. Telecommuting permitted and Travel may be required to various unanticipated locations throughout the United States.

SAN JOSE/MILPITAS/SANTA CLARA, CA: IT Engineer (Ref.# 7): Responsible for development, support and implementation of major system functionality of company's proprietary networking products. **Engineering Architect (Ref.#189):** Lead Business Intelligence (BI) operations and applications through Extraction Transformation and Loading (ETL) processes and system engineering and analysis. **Technical Marketing Engineer (Ref.# SJ15):** Responsible for enlarging company's market and increasing revenue by marketing, supporting, and promoting company's technology to customers. **Technical Marketing Engineer (Ref.# SJ178):** Responsible for enlarging company's market and increasing revenue by marketing, supporting, and promoting company's technology to customers. Travel may be required to various unanticipated locations throughout the United States. **Technical Lead/Leader (Ref.# SJ658):** Lead engineering groups on projects to design, develop or test hardware or software products. Telecommuting permitted. **Diagnostic Engineer (Ref.# SJ168):** Design and develop diagnostic software for verification and validation in engineering and manufacturing. **Network Engineer (Ref.# SJ575):** Responsible for the operational support of internal network systems. **Customer Support Engineer (Ref.# SJ3):** Responsible for providing technical support regarding the company's proprietary systems and software. **IT Manager (Ref.# SJ182):** Responsible for small, medium, large/complex and multiple projects throughout the project lifecycle (initiate, plan, execute, control, close) or a portion of a larger, more complex project.

PLEASE MAIL RESUMES WITH REFERENCE NUMBER TO CISCO SYSTEMS, INC., ATTN: M51H, 170 W. Tasman Drive, Mail Stop: SJC 5/1/4, San Jose, CA 95134. No phone calls please. Must be legally authorized to work in the U.S. without sponsorship. EOE.

www.cisco.com

 2469-7087/16/$33.00 © 2016 IEEE

workarounds or solutions for customers; Determining root cause of tech problems by reviewing product source code using traces, dumps, & debuggers. Send resume to: Althea Wilson, CA Technologies, One CA Plaza, Islandia, NY 11749, Refer to Requisition #123743

COMPUTER PROGRAMMERS: Write, update, & maintain computer programs, review programs. Requirements Gathering, Data Mapping, Data Conversion, Report Generation, SDLC methodologies, Version Control Tools, Mainframe,XML, Unix, SQL Server, Oracle, Java/J2ee/Spring/Struts/Websphere/Apache, Waterfall/Agile methodologies. Will work in unanticipated locations. Req. 2 yrs experience. Send resume to NobleSoft Solutions, Inc. 405 Executive Dr. Langhorne, PA 19047

SR. PROGRAMMER sought by Ratan Palace LLC, North Bergen, NJ. Req. MS or equiv in Comp. Prgmg or any rltd field to support hotel info processing needs. Resume Attn: Nehal Gandhi, Ratan Palace LLC, Holiday Inn Express, 2600 Tonnelle Ave, North Bergen, NJ 07047, or email: HIEXPRESSNB@GMAIL.COM

Oracle America, Inc.

has openings in **Portland, OR** for:

SOFTWARE DEVELOPERS

Job duties include:

Design, develop, troubleshoot and/or test/QA software. May telecommute from home.

Apply by e-mailing resume to cecilia.broadaway@oracle.com referencing 385.16954.

Oracle supports workforce diversity.

JUNIPER NETWORKS

Juniper Networks is recruiting for our Sunnyvale, CA office:

Strategic Delivery Manager, Sr. Staff #35314: Build a strong engagement and interlock between Juniper's engineering organization and customers' planning organization to drive evolution of key platforms, architectures and solutions.

Software Engineer #37019: Design, develop and debug kernel OS features for JUNOS. Work on high-availability features and sustaining existing kernel features. Support customer deployments and escalations.

UI Design Engineer Staff #31689: Architect, design, and implement UI development for all Cloud Service Platform (CSP) applications. Work closely with UI development team to implement and/or assist in implementing user interface elements. Architect and design UI automation framework.

Solution Consultant Staff #20095: Be a subject matter expert on Company's security products in order to develop and deliver detailed solutions and implement products for customers. May work at other undetermined locations throughout the U.S. Telecommuting allowed.

ASIC Engineer #35272: Define, develop, architect, code, modify, evaluate and deliver verification suites and tests for Application- Specific Integrated Circuits (ASICs).

Software Engineer Senior Staff #17486: Plan and write detailed design documents for large software systems, and develop complex software programs in C, Python and TCL.

Software Engineer #30134: Architect, design and develop Linux and FreeBSD based device drivers. Develop embedded applications to manage various platform components (ASICs, temp/voltage monitors) using C and C++ object oriented design.

Test Engineer #10840: Develop, modify, and execute testing methods for Routing protocols and SDN for upcoming new features and functionality in Junos (Company's routers operating system).

Sustaining Engineer #35053: Troubleshoot, debug, and replicate complex and unique technical issues. Analyze customer issues escalated from support engineering.

Technical Support Engineer #7108: Provide technical support and high level expertise to customers on Company products.

Business Operations Analyst #33416: Analyze complex business problems and issues by leveraging prior training in data mining, SQL databases, and data science (Python, R, visualization tools) to establish company-wide big data analytics.

Juniper Networks is recruiting for our Westford, MA office:

Technical Support Engineer #36011: Support Secure Routing products, working directly with customers and partners. Work with highly knowledgeable group of customers and act as an escalation point for other TAC groups within the organization.

Mail single-sided resume with job code # to
Juniper Networks
Attn: MS A.8.429A
1133 Innovation Way
Sunnyvale, CA 94089

TEMPE, AZ

Premier Field Engineer, Platforms: Provide technical support to enterprise customers, partners, internal staff or others on mission critical issues experienced w/ Microsoft technologies. Requires travel up to 75% with work to be performed at various unknown worksites throughout the U.S. Telecommuting permitted. https://jobs-microsoft.icims.com/jobs/4789/go/job

Mountain View, Palo Alto, Sunnyvale, CA

Applied Scientist: Utilize knowledge in applied statistics and mathematics to handle large amounts of data using various tools. http://bit.ly/MSJobs_Data_Applied_Science

Design Verification/Validation Engineers: Responsible for ensuring the quality of Microsoft hardware products. http://bit.ly/MSJobs_Hardware_Design_Verification_Eng

Designers: Develop user interface and user interaction designs, prototypes and/or concepts for business productivity, entertainment or other software or hardware applications. http://bit.ly/MSJobs_Design

Hardware Dev., Test or Design Engineers, Hardware Engineers, Electrical Engineers, Design Engineers (all levels, including Leads and Managers): Design, implement and test computer hardware that add strategic value to the company. (http://bit.ly/MSJobs_Hardware_Dev_Eng)(http://bit.ly/MSJobs_Electrical_Eng)

Hardware Dev., Test or Design Engineers, Hardware Engineers, Electrical Engineers, Design Engineers (all levels, including Leads and Managers): Design, implement and test computer hardware. Requires Domestic and International travel up to 25%. (http://bit.ly/MSJobs_Hardware_Dev_Eng)(http://bit.ly/MSJobs_Electrical_Eng)

Premier Field Engineers: Provide technical support to enterprise customers, partners, internal staff or others on mission critical issues experienced with Microsoft technologies. Roving Employee—requires travel up to 100% with work to be performed at various unknown worksites throughout the U.S. Telecommuting permitted. http://bit.ly/MSJobs_Support_Delivery

Researchers/Scientists: Conduct research and lead research collaborations that yield new insights, theories, analyses, data, algorithms, and prototypes and that advance state-of-the-art of computer science and engineering, as well as general scientific knowledge. http://bit.ly/MSJobs_Research

Solution Managers: Identify and analyze internal client and partner business needs, and translate needs into business requirements and value-added solutions and solution roadmaps. http://bit.ly/MSJobs_IT_Solution_Mgmt

MOUNTAIN VIEW

Technology Solutions Professional: Engage w/ strategic customers to solve their business needs by conceiving, designing, & implementing transformative technical architecture. Requires domestic travel up to 25%. https://jobs-microsoft.icims.com/jobs/4812/job

PALO ALTO, CA

Designer II: Develop user interface & user interaction designs, prototypes &/or concepts for business productivity, entertainment or other software or hardware applications. Requires dom. and intl. travel up to 25%. https://jobs-microsoft.icims.com/jobs/4858/job

FARGO, ND

Technical Advisor: Provide technical advice & support on issues experienced w/ Microsoft technologies. Requires dom. travel up to 25%. https://jobs-microsoft.icims.com/jobs/4842/job

CHARLOTTE, NC

Technical Account Managers: Assure productive use of Microsoft technologies, focusing on delivery quality through planning and governance. http://bit.ly/MSJobs_Delivery_Relationship_Mgmt

Senior Consultant: Deliver design, planning, & implementation services that provide IT solutions to customers & partners. Dom. & intl. travel up to 75%. Telecommuting permitted. https://jobs-microsoft.icims.com/jobs/4814/job

Technical Advisor: Support commercial SharePoint across the online services go-to-markets. https://jobs-microsoft.icims.com/jobs/4780/go/job

MAHWAH, NJ

Support Engineers / Escalation Engineers: Install, configure, support and troubleshoot issues related to Microsoft technologies. http://bit.ly/MSJobs_Support_Eng

NEW YORK, NY

Consultants: Deliver design, planning, and implementation services that provide IT solutions to customers and partners. Requires domestic and international travel up to 25%. http://bit.ly/MSJobs_Technical_Delivery

Data Scientist: Manipulate large volumes of data, create new and improved techniques and/or solutions for data collection, management and usage. http://bit.ly/MSJobs_Data_Applied_Science

Designers: Develop user interface and user interaction designs, prototypes and/or concepts for business productivity, entertainment or other software or hardware applications. http://bit.ly/MSJobs_Design

Solution Architect: Architect software, platform, services, hardware or technology solutions. Requires travel up to 25% w/ work to be performed at various unknown worksites throughout the U.S. Telecommuting permitted. https://jobs-microsoft.icims.com/jobs/4791/go/job

Technical Evangelist: Collaborate w/ sales teams to understand customer requirements, promote the sale of products, & provide sales support. Requires travel up to 50% w/ work to be performed at various unknown worksites throughout the U.S. https://jobs-microsoft.icims.com/jobs/4767/job

IRVING, TX

Service Engineers/Managers, Service Operations Engineers, and Systems/Operations Engineers/Site Reliability Engineer: Research, design, develop, and test operating systems-level software, compilers, and network distribution software. (http://bit.ly/MSJobs_Service_Engineering) (http://bit.ly/MSJobs_IT_Serv_Eng)(http://bit.ly/MSJobs_IT_Serv_Ops)

Support Engineers / Escalation Engineers: Install, configure, support and troubleshoot issues related to Microsoft technologies. http://bit.ly/MSJobs_Support_Eng

Premier Field Engineer, SharePoint: Provide technical support to enterprise customers, partners, internal staff or others on mission critical issues experienced w/ Microsoft technologies. Requires domestic travel up to 50%. https://jobs-microsoft.icims.com/jobs/4851/job

Support Escalation Engineer: Install, configure, support and troubleshoot issues related to Microsoft technologies. Telecommuting permitted. https://jobs-microsoft.icims.com/jobs/4925/job

HOUSTON, TX

Technical Account Manager: Assure productive use of Microsoft technologies, focusing on delivery quality through planning & governance. Requires travel up to 25% throughout the U.S. https://jobs-microsoft.icims.com/jobs/4870/go/job

IRVINE, CA

Account Technology Strategist: Identify & analyze internal clients & partner business needs, and translate needs into business requirements & value-added solutions & solution roadmaps. Requires travel up to 25% w/ work to be performed at various worksites throughout the U.S. https://jobs-microsoft.icims.com/jobs/4773/job

Technology Solutions Professional: Improve the Enterprise Mobility business metrics (revenue & scorecard) through excellence in technical sales strategy & execution. Requires travel throughout U.S. up to 50% w/ work to be performed at various unknown worksites through the U.S. Telecommuting permitted. https://jobs-microsoft.icims.com/jobs/4771/job

Premier Field Engineer, SQL: Provide technical support to enterprise customers, partners, internal staff or others on mission critical issues experienced w/ Microsoft technologies.

Multiple job openings are available for each of these categories. To view detailed job descriptions and minimum requirements, and to apply, visit the website address listed. EOE.

Microsoft Corporation currently has the following openings (job opportunities available at all levels, including Principal, Senior and Lead levels):

REDMOND, WA

Business Program Managers: Plan, initiate, and manage technology and business projects. http://bit.ly/MSJobs-Buss_Oper_Prog_Mgmt

Operations Program Managers: Plan, initiate, and manage information technology (IT) projects. (http://bit.ly/MSJobs_Ops_PM)

Program Managers: (All levels, including Leads and Managers) Coordinate program development of computer software applications, systems or services working with development and product planning teams. Requires Domestic and International travel up to 25%. (http://bit.ly/MSJobs_ProgMgr)(http://bit.ly/MSJobs_HW_ProgMgr)(http://bit.ly/MSJobs_ProdQlty_Supp)(http://bit.ly/MSJobs_IT_ProgMgr)

Program Managers: (All levels, including Leads and Managers) Coordinate program development of computer software applications, systems or services working with development and product planning teams. (http://bit.ly/MSJobs_ProgMgr) (http://bit.ly/MSJobs_HW_ProgMgr)(http://bit.ly/MSJobs_ProdQlty_Supp)(http://bit.ly/MSJobs_IT_ProgMgr)

Business Program Manager: Plan, initiate, & manage technology & business projects. Telecommuting Permitted. Position allows employee to reside anywhere in the U.S. and telecommute to perform work exclusively from home. https://jobs-microsoft.icims.com/jobs/4875/go/job

Business Program Managers: Plan, initiate, & manage technology and business projects. Requires dom. and intl. travel up to 25%. https://jobs-microsoft.icims.com/jobs/4878/job?mode=view; https://jobs-microsoft.icims.com/jobs/4867/go/job

Hardware Engineering Program Manager: Coordinate program development of hardware products or systems, working w/ development & product planning teams. Requires dom., regional & intl. travel up to 50%. https://jobs-microsoft.icims.com/jobs/4873/go/job

Hardware Engineering Program Manager: Coordinate program development of computer hardware products, working w/ engineering & product sourcing teams. https://jobs-microsoft.icims.com/jobs/4753/go/job

Program Manager, NPI: Coordinate program development of hardware products or systems, working w/ development & product planning teams. Requires dom., regional & intl. travel up to 50%. https://jobs-microsoft.icims.com/jobs/4869/go/job

Sr. Program Manager: Coordinate program development of computer software applications, systems or services, working with development and product planning teams. Telecommuting permitted. https://jobs-microsoft.icims.com/jobs/5450/job

FT. LAUDERDALE, FL

Operations Program Managers: Plan, initiate, and manage information technology (IT) projects. (http://bit.ly/MSJobs_Ops_PM)

RENO, NV

Operations Program Managers: Plan, initiate, and manage information technology (IT) projects. (http://bit.ly/MSJobs_Ops_PM)

MOUNTAIN VIEW, PALO ALTO, SUNNYVALE, CA

Program Managers: (All levels, including Leads and Managers) Coordinate program development of computer software applications, systems or services working with development and product planning teams. (http://bit.ly/MSJobs_ProgMgr) (http://bit.ly/MSJobs_HW_ProgMgr)(http://bit.ly/MSJobs_ProdQlty_Supp)(http://bit.ly/MSJobs_IT_ProgMgr)

SAN FRANCISCO, CA

Program Managers: (All levels, including Leads and Managers) Coordinate program development of computer software applications, systems or services working with development and product planning teams. (http://bit.ly/MSJobs_ProgMgr) (http://bit.ly/MSJobs_HW_ProgMgr)(http://bit.ly/MSJobs_ProdQlty_Supp)(http://bit.ly/MSJobs_IT_ProgMgr)

Multiple job openings are available for each of these categories. To view detailed job descriptions and minimum requirements, and to apply, visit the website address listed. EOE.

understand customer requirements, & transfer acquired knowledge into a technical plan of action. Travel Required. Mail resume w/job code #36514 to: Cloudera, Attn.: HR, 1001 Page Mill Rd., Bldg. 2, Palo Alto, CA 94304.

SPLUNK INC. has the following job opportunities in San Francisco, CA: **Senior Product Specialist (Product Lifecycle Engineer) (REQ#9YF3N8).** Dev prod reqs, strategy & roadmap for Co's Fraud & Account Takeover prods. 20% travel req. **Software Engineer (REQ#9RV2RR).** Design & dev new features for Co.'s core search & analytics interface. Splunk Inc. has the following job opportunities in Palo Alto, CA: **Principal Software Engineer (REQ#9Z5VY9).** Architect, design & develop Co.'s ETL engine, data platform, connectors, parsers, & associated logic. **Data Scientist (REQ#9Z5W24).** Design & dev cloud infrastructure & services for threat intelligence. Splunk Inc. has the following job opportunity in Cupertino, CA: **Senior CRM Business Analyst (REQ#9JVVGD).** Coord with business process owners, ERP integration architects, & business reps to gather & document business requirements for key salesforce.com initiatives. Refer to Req# & mail resume to Splunk Inc., ATTN: J. Aldax, 250 Brannan Street, San Francisco CA 94107. Individuals seeking employment at Splunk are considered without regards to race, religion, color, national origin, ancestry, sex, gender, gender identity, gender expression, sexual orientation, marital status, age, physical or mental disability or medical condition (except where physical fitness is a valid occupational qualification), genetic information, veteran status, or any other consideration made unlawful by federal, state or local laws. To review US DOL's EEO is The Law notice please visit: https://careers.jobvite.com/Splunk/EEO_poster.pdf. To review Splunk's EEO Policy Statement please visit: http://careers.jobvite.com/Careers/Splunk/EEO-Policy-Statement.pdf. Pursuant to the San Francisco Fair Chance Ordinance, we will consider for employment qualified applicants with arrest and conviction records.

ASSISTANT/ASSOCIATE/FULL RESEARCH PROFESSOR IN CYBER SECURITY. KINDI Center For Computing Research / Qatar University. Qatar University invites applications for research faculty positions at all levels. Candidates will cultivate and lead research projects at the KINDI Center for Computing Research in the area of Cyber Security. Qatar University offers competitive benefits package including a 3-year renewable contract, tax free salary, free furnished accommodation, and more. Apply by posting your application on the QU online recruitment system at careers.qu.edu.qa under "College of Engineering"

EXPEDIA, INC. currently has openings for the following opportunities in our Bellevue, WA office (various/levels/

to design technical solutions and provide detailed specifications that satisfy scalability, reliability, and performance goals. Managers, Software Engineering (728.868) Manage Agile software development phases of initiatives across Hotwire's travel platform. Specifically, ensure proper technical design of applications and quality standards are upheld. Send your resume to: Hotwire/Expedia Recruiting, 333 108th Avenue NE, Bellevue, WA 98004. Must reference position and Job & Job ID# listed above.

CARRENTALS.COM, INC. currently has openings for the following opportunities in our San Francisco, CA office (various/levels/types:) Software Engineers: (728.SWE-APR-CAR) Design, implement, and debug software for computers including algorithms and data structures. Send your resume to: CarRentals/Expedia Recruiting, 333 108th Avenue NE, Bellevue, WA 98004. Must reference position and Job & Job ID# listed above.

ORBITZ WORLDWIDE LLC currently has openings for the following opportunities in our Chicago, IL office (various/levels/types:) Software Engineers: (728.SWE-APR-ORB) Design, implement, and debug software for computers including algorithms and data structures. Send your resume to: Orbitz/Expedia Recruiting, 333 108th Avenue NE, Bellevue, WA 98004. Must reference position and Job & Job ID# listed above.

COMPUTERS: TECH MAHINDRA (AMERICAS) INC. is seeking to fill numerous IT positions. Prgrm Mgrs to oversee & manage mult. IT projects, proj. planning, dvlpmnt, implementation, acct & delivery mgmt; Proj. mgrs to oversee & manage IT teams w/dvlpmnt of various sftwre apps. Sys. Analyst/Programmers/Quality & Tech Architect/ Tech Solutions Architect /Quality Engg /Test Eng/ Sftwre Eng. / sftwre Developer / Systems Admin/DBAs/DB Architect /Network Engr/Network Admin/ to analyze, design, dvlp, test & maintain comp software apps, systems, databases, or networks through all phases of sftwre dvlpmt life cycle (Sftwre Eng. may also lead a team on various projects); Telecom solutions Architect / Application Architects/Electrical Eng to use various technologies to dsgn & dvlop telecom software apps. Mech. Eng (CAD/CAM) to design, develop, validate & perform structural calculations, product improvement, & provide tech. support to design teams at high levels utilizing specific mech. tools. Sales Eng/Bus. Analyst/Mgmt Analyst for solutions/pre-sales activities w/relev. industry experience. IT Bus. Dev. Mgrs to create new business, negotiate contracts & dvlp proposals for customized IT solutions. Rel. Mgrs to manage/outsource commercial IT/Eng. deals, monitor & maintain existing accts. All Tech/Mgrial. positions require a MS or BS degree or foreign equiv. in CS, Comp Apps, CIS, IT, Eng, Bus mgmt/admin or closely related fields and relevant industry exp. Sales/Rel.Mgrs. require a MS or BS degree or foreign equiv. in Bus. Admin, Eng. or closely related field & relev. Industry exp. Positions are based out of corp. HQ in 4965 Preston Park Blvd, # 500, Plano, TX 75093 & subject to travel & relocation to client sites located throughout the U.S. Mail resume & position applied for w/JOB CODE: 04IE16 to Visa Cell, Tech Mahindra (Americas) Inc., 1001 Durham Avenue, Suite 101, South Plainfield NJ 07080

MANAGER. Job location: Miami, FL & any other unanticipated locations in U.S. Travel Required. Duties: Lead Oracle DRM implement. to support Lawson ERP with Hyperion Financial Mgmt., EPM Architect & Essbase. Architect Oracle DRM solutions to define & support master data mgmt.. & governance process for clients. Design & develop integrations between ERP systems incl. Lawson, Oracle, EBS, PeopleSoft, Business Intelligence Appls.

中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY ,CHINESE ACADEMY OF SCIENCES

Institute of Computing Technology, Chinese Academy of Sciences Faculty Positions

The Institute of Computing Technology of the Chinese Academy of Sciences (hereafter ICT) is the cradle for China's computer profession, having spun off many academic institutions and hi-tech companies. They include the Institute of Microelectronics, Computing Center, Institute of Software Computer Network Information Center, all of the Chinese Academy of Sciences, as well as hi-tech companies such as Lenovo and Dawning.

ICT holds firmly to the new-era guidelines of the Chinese Academy of Sciences, focusing on fundamental, strategic and advanced research. It aims to becoming a leader in computing technology, being concerned with the national interest and people's livelihood, and leading Chinese IT industry.

For more information, visit www.ict.ac.cn.

ICT is seeking candidates for faculty position in Computing technology areas—including but not limited to high performance computer, Big data and social computing, Computer architecture, New computer system, Integrated circuit design, Compiling and programming, Intelligent perception calculation, Multidisciplinary researches on life science and social science, Intelligent information processing.

Qualifications:

Ph. D in Computer Science/ Engineering, Electrical Engineering, Communication, or closely related field;

Compensation and Benefits:

1) The salary and benefit package are highly competitive, commensurate with experience and academic accomplishment.

2) The candidates with strong background are eligible for applying for"1,000 Talents Plan(China)""100 Talents Program(CAS)".

Application Materials:

CV; supporting documents such as certifying your highest degree, academic achievement/awards, and the current employment etc; 3 representative papers.

Contact information

hr@ict.ac.cn, Phone:86-10-62601207

CLASSIFIED LINE AD SUBMISSION DETAILS: Rates are $425.00 per column inch ($640 minimum). Eight lines per column inch and average five typeset words per line. Send copy at least one month prior to publication date to: Debbie Sims, Classified Advertising, *Computer* Magazine, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720; (714) 816-2138; fax (714) 821-4010. Email: dsims@computer.org.

In order to conform to the Age Discrimination in Employment Act and to discourage age discrimination, *Computer* may reject any advertisement containing any of these phrases or similar ones: "…recent college grads…," "…1–4 years maximum experience…," "…up to 5 years experience," or "…10 years maximum experience." *Computer* reserves the right to append to any advertisement without specific notice to the advertiser. Experience ranges are suggested minimum requirements, not maximums. *Computer* assumes that since advertisers have been notified of this policy in advance, they agree that any experience requirements, whether stated as ranges or otherwise, will be construed by the reader as minimum requirements only. *Computer* encourages employers to offer salaries that are competitive, but occasionally a salary may be offered that is significantly below currently acceptable levels. In such cases the reader may wish to inquire of the employer whether extenuating circumstances apply.

Incl. OBIEE, Essbase, Hyperion Planning, Hyperion Financial Mgmt. & DRM to manage & govern complete Chart of Accounts & reporting. Develop, design, build & support all DRM objects & integrations on Windows & Unix environ. using Oracle database & MS SQL Server. Requires: Master's degree in Comp. Sci, Bus. Adm., Eng. or related field & 3 yrs. exp. in job offered or 3 yrs. exp. as a Consultant or Systems Analyst. Will accept B.S. (or foreign equiv.) & 5 yrs. exp. in the comp. ind. in lieu of M.S. & 3 yrs. exp. Concurrent exp. must incl.: 3 yrs. exp. with Hyperion DRM & Essbase & 1 yr. exp. leading Oracle DRM implementations. Send resume (no calls) to: Michelle Ramirez, The Hackett Group, Inc., 1001 Brickell Bay Dr., Suite 3000, Miami, FL 33131.

DATA WAREHOUSE DESIGNERS MULTIPLE POSITIONS AVAILABLE IN ENFIELD, CT. Analyze and translate complex designs into unit tested and maintainable code. Design components of an application, the overall operating system, or complex sub-systems such as sophisticated file management routines. Design and develop new features and applications using Data Warehouse and Extract Transform and Load (ETL) tools including Teradata, Informatica, and Software Development Life Cycle (SDLC). Apply: L Sawtelle, Massachusetts Mutual Life Insurance Company, 1295 State Street, Springfield, MA 01111; Please Reference Job ID: 708202200

ELECTRICAL GEODESICS, INC. seeks to hire a Global Support Scientist for its office in Eugene, Oregon. Define Software Requirement Statements in biotechnology, including EGI technology; communicate research findings and present on EGI products; analyze technology needs; provide input to improve EGI's biotechnology. Mail resume to: Electrical Geodesics, Inc., 500 E. 4th Avenue, Suite 200, Eugene, OR 97401. Attn: Carrie Baltzer

MANAGER. Job location: Miami, FL & any other unanticipated locations in U.S. Travel Required. Duties: Manage client engagements incl. creating work plans detailing project tasks, milestones, duration, & resource allocation for financial services clients using MS Project. Develop effort estimations for project tasks using WBS (work breakdown structure), & size based estimation approaches. Coord. with offsite & onsite teams int'l for delivering IT services. Interface with clients for solution implement. activities incl. leading requirements definition using tools such as PowerPoint & Excel, & testing & defect mgmt. activities using tools such as Qual. Ctr., Windows Test Tech. (WTT), & Product Studio. Create process flow diagrams with MS Visio for documenting current state processes & showing future state recommends. Analyze opps. for optimizing IT cost by reviewing oper. processes & related appls. Present recommends. & related roadmaps using tools such as Excel, PowerPoint & Visio. Requires: Master's degree in Comp. Sci, Bus. Adm., Eng. or related field & 3 yrs. exp. in job offered or 3 yrs. exp. as a Sr. Consultant, Software Eng., Prog/Analyst or Specialist. Will accept B.S. (or foreign equiv.) & 5 yrs. exp. in the comp. ind. in lieu of M.S. & 3 yrs. exp. Concurrent exp. must incl.: 3 yrs. exp. creating process flow diagrams with MS Visio; 3 yrs. exp. presenting recommends. & roadmaps; & 3 yrs. exp. using WBS, WTT & Product Studio. Send resume (no calls) to: Michelle Ramirez, The Hackett Group, Inc., 1001 Brickell Bay Dr., Suite 3000, Miami, FL 33131

TECHNICAL

Oracle America, Inc.

has openings for

TECHNICAL ANALYST

positions in **Lehi, UT**.

Job duties include: Analyze user requirements to develop, implement, and/or support Oracle's global infrastructure. As a member of the IT organization, assist with the design, development, modifications, debugging, and evaluation of programs for use in internal systems within a specific function area.

Apply by e-mailing resume to paul.merideth@oracle.com, referencing 385.19389.

Oracle supports workforce diversity.

TECHNOLOGY

LinkedIn Corp.

has openings in our **Mtn View, CA** location for:

Senior Information Security Manager (6597.980)

Lead a team that is responsible for protecting infrastructure, applications, & members — specifically identifying new vulnerabilities & responding to existing vulnerabilities within the organization.

LinkedIn Corp. has openings in our **San Francisco, CA** location for:

Senior Software Engineer (6597.1602)

Design, develop & integrate cutting-edge software technologies.

Please email resume to: 6597@linkedin.com. Must ref. job code above when applying.

TECHNICAL

Oracle America, Inc.

has openings for

TECHNICAL ANALYST

positions in **Lehi, UT**.

Job duties include: Analyze user requirements to develop, implement, and/or support Oracle's global infrastructure.

Apply by e-mailing resume to dustin.halliday@oracle.com, referencing 385.19774.

Oracle supports workforce diversity.

HW. Travel req'd 25%.

Software Development Engineer (Req#9WXRFU) Respons for app life cycle from prototyping to release to mass mrkt as well as road map & release planning of user facing app, includ prototyping dsgn, dev, tstng & doc of custom-made solutions.

Software Engineer Systems (Req#9X-4UAQ). Res for dvlp video codec firmware, drivers & algrthm for current & future prdcts.

Software Engineer Applications (Req#9RU43Q). Dev new & existing Android apps for consumer media.

ASIC Design Engineer (Req#-9CYTXN). Design test logics & generate test patterns for app processors.

Software Development Engineer (Req#9K2NQ8). Des & dev tools, automation, and tsting infrastruct for iOS Accessories.

Software Development Engineer (Req#9ZR23U). Research, des, dev, impl & debug embedded SW for wireless ntwrking in next gen prods

Hardware Development Engineer (Req#9MT2B2). Des, dev, tst, & supervise the manufact of high-perf camera lenses for mobile dev. Travel req 20%

Hardware Development Engineer (Req#9PQTW3). Des & valid optical sensor sys for consumer electronic products. Travel req 15%.

Software Engineer Systems (Req#9W-9SUS) Des, dev & config netwrk'g sys HW and SW.

Information Systems Engineer (Req#9UQT58) Provide sol, sup & main on infrastruct of the integrat plat rel to supply chain logis & legal.

Product Design Engineer (Req#9QT-SUZ). Des & dev HW interconn solutions.

Software Development Engineer (Req#9DCT5J). Contribute to current & future app tech.

Software Engineer Applications (Req#9P92GL) Build SW & systms to bring about infrastructure & apps through automation.

Network Systems Manager (Req#9D-NUU7). Resp for network ops within Apple's global data centers.

Mechanical Design Engineer (Req#9A63KF) Set priorities & determine risk mitigation plans using an understanding of techn details of cross-func issues & risks in mnfctring. Travel req: 30%.

Software Development Engineer (Req#9ZA64R) Res for the end to end quality of the Thai Siri exp. Prfrm tech device tests of Siri in Thai.

Software Quality Assurance Engineer (Req#9YGUBV) Tst full agile life cycle SW of MS Office & iWork Cmptblty for OSX, iOS, &iCloud pltfrms.

Software Development Engineer (Req#9N7T2Q) Contrbte & bld Apple's future wirelss prdcts by tsting Bluetooth for iOS devices. Travel req 15%.

Software Development Engineer (Req#9NKTDB) Resp for the sgn & dvlpmnt of SW for the Siri srvr pltfrm.

Engineering Project Lead (Req#9T2TZ5) Dev & plan review of proj spec's for new maps geo-special feats for mobile dev's in area of points of interest & businesses.

Business Systems Analyst (Req#9JF-PVH). Resp for complex, global initiatives in procurement & materials

Software Engineer Applications (Req#9QY2TG). Gather bus reqs, provide GAP analysis & complete code des and dev for the MFi App & the integ partners of the MFi App

Software Engineer Applications (Req#9Q3RZY). Des, dev, impl, & maintain web services for the Apple build team

Software Engineer Applications (Req#9VNPZL). Dev server SW for device activation & enablement.

Software Engineer Applications (Req#9EP3KS) Design & dev distributed sw to power iCloud apps & srvcs.

Engineering Project Lead (Req#9T-V3ZG). Work with dsign & cross functn team to select, evaluate & qualf new supplrs. Travel req'd 30%.

Software Development Engineer (Req#9FH2TH) Dsgn & dev a digital personal assist for mobile devices.

Apple Inc. has the following job opportunity in Austin, TX:

ASIC Design Engineer (Req#9F7QCM) Resp for the pre-silicon RTL verifictn of to level embedded graphics cores

Refer to Req# & mail resume to Apple Inc., ATTN: L.J., 1 Infinite Loop 104-1GM, Cupertino, CA 95014. Apple is an EOE/AA m/f/disability/vets.

Apple Inc. has the following job opportunity in Cupertino, CA:

Production Designer (Req#9K5Q88) Compose elegant, det digital layouts for web, email comm, in-store interactive kiosks, and other proj with art direction from the IC Team.

Interested applicants must submit a portfolio that demonstrates skills required. Please enclose a self-addressed stamped envelope if you wish your portfolio to be returned. Refer to Req# & mail resume to Apple Inc., ATTN: L.J., 1 Infinite Loop 104-1GM, Cupertino, CA 95014. Apple is an EOE/AA m/f/disability/vets.

TECHNOLOGY

SanDisk Corporation

has openings for the following positions in **Milpitas, California**:

Design Engineers (Job code: SD-DS316): Design, develop, modify and evaluate electronic parts, components or integrated circuitry for electronic equipment and other hardware systems. Determine design approaches and parameters. **Device Engineers (Job code: SD-DV316):** Design, develop, modify and evaluate digital electronic parts, components or integrated circuitry for digital electronic equipment and other hardware systems. Determine design approaches and parameters. **Product Engineers (Job code: SD-PD316):** Provide production support engineering for a specific product or group of products after transfer from design to high volume production. Interface with design, process, test and reliability engineering to solve problems. **Firmware Engineers (Job code: SD-FW316):** Analyze, design, program, debug, troubleshoot and/or modify software for firmware (IC embedded code) applications. Conduct programming activities involving analog and digital hardware and software operating systems. **Industrial & Operations Engineers (Job code: SD-IO316):** Responsible for the development of estimated or engineering standards, optimum process layouts and methods. Respond to shifting priorities and business needs by identifying, developing, and promoting new technologies, methods and processes. **Verification Engineers (Job code: SD934):** Participate in the logic design and verification of memory components and develop test bench architecture. **Systems Design Engineers (Job code: SD578):** Design, define and implement complex system requirements for customers and prepare studies and analyze existing systems. Some travel may be required to work on projects at various, unanticipated sites throughout the United States. **Managers I, Supply Chain Optimization (Job code: SD127):** Responsible for design, development and implementation of supply chain process improvements and system changes. **Verification Engineers (Job code: SD269):** Perform Standard Cell Library development involving schematic development, verification, characterization, model development and library delivery. **Software Engineers (Job code: SD69):** Define, implement, and execute benchmark testing to meeting product requirements while working within NAND memory performance limitations. **System Design Engineers (Job code: SD213):** Validate features including writing up a test plan and cross referencing it to individual requirements. **Senior Staff Process Integration Engineers (Job code: SD467):** Provide production support engineering for NAND product after transfer from design to high volume production. Must be available to work on projects at various, unanticipated sites throughout the United States and abroad. **Business Systems Analysts (Job code: SD385):** Understand technology infrastructure and build workable web and mobile solutions. Some travel may be required to work on projects at various, unanticipated sites throughout the United States. **Technical Rotation Engineers (Job code: SD471):** Responsible for working closely with several cross functional teams related to design, device, firmware, test, marketing, quality and manufacturing teams to define/develop/maintain a product test flow.

Positions in **San Jose, California**:

Senior Software Development Engineers (Job code: SD433): Develop software for application accelerating solutions. May telecommute from home.

To apply, reference job code # & mail resume to: SanDisk Corporation, 951 SanDisk Drive, MS: HRGM, Milpitas, CA 95035. EOE

Archaeology by Internet

by Charles Day
American Institute of Physics
cday@aip.org

One day in March 2007, economist Dmitry Dey was watching a Discovery Channel documentary about pyramids, mummies, and tombs at his home in Kostanay, a city of 217,000, in northern Kazakhstan. An enthusiast of archaeology, Dey speculated that long-forgotten pyramids might also have risen from the flat, empty steppe around Kostanay. To satisfy his curiosity, he trawled satellite imagery via Google Earth.

As *New York Times* reporter Ralph Blumenthal recounted in a news story he wrote last year, Dey didn't find pyramids. Instead, he discovered several ancient earthworks. The largest—known as the Ushtogay Square—consists of 110 mounds arranged in a 286×286 m^2 square with diagonals connecting the corners.

Since Dey's discovery, archaeologists have visited Ushtogay Square and other sites. Given the absence of human remains, the structures don't appear to be tombs. Artifacts found nearby suggested to Dey a link to the nomadic Mahandzhars, who herded livestock in the region from 7000 BCE to 5000 BCE. However, when a technique known as optically stimulated luminescence was used to date a sample of mound material, it yielded an age of 2,800 years.

Like Dey, I was curious about the earthworks. Unlike him, I turned instead to Google Scholar. Surprisingly, I could find just two papers that mentioned the Mahandzhars. Wikipedia has no entry on the culture, nor does the 3,048-page 1993 edition of the *Columbia Encyclopedia*. Yet knowing who built the earthworks wouldn't answer the bigger questions of why and how they did it.

Could computation provide answers? Possibly. Google Scholar also led me to a preprint by Carsten Lemmen and Kai Wirtz of the Institute of Coastal Research in Geesthacht, Germany. The two ecosystem modelers set themselves the task of determining whether a change in climate played a significant role in the spread of farming to Europe, which started in the Fertile Crescent around 10,000 BCE and ended in Scandinavia around 3500 BCE.

As you might expect, Lemmen and Wirtz used paleoclimatological methods to determine the past climate. To ascertain how a network of Neolithic European tribes might change its socioeconomic behavior in response to changes in climate, technology, and other factors, they used their Global Land Use and technological Evolution Simulator (GLUES). In the words of their paper, GLUES "mathematically resolves the dynamics of local human populations' density and characteristic sociocultural traits in the context of a changing biogeographical environment."

After running GLUES with the ancient European climate, Lemmen and Wirtz could reproduce the timing and distribution of the westward spread of farming as deduced from archaeological evidence. And they could do so without having to invoke climate change as a facilitating factor.

Now that Lemmen and Wirtz have validated their approach in Europe, it would be fascinating to see what they might find if they applied it to the same time period in Central Asia. When might one of the region's tribes have reached the socioeconomic maturity to build the likes of an Ushtogay Square?

Looking instead to humanity's future in space, GLUES and simulators like it could one day be used to predict whether a modest colony established on a distant moon would spread and thrive or shrink and wither. ■

Charles Day is *Physics Today*'s editor in chief. The views in this column are his own and not those of either *Physics Today* or its publisher, the American Institute of Physics.

This article originally appeared in Computing in Science & Engineering, *vol. 18, no. 2, 2016.*

