COMPUTING edge

> Internet of Things
> Social Media
> Cyber-Physical Systems
> Wearables

OCTOBER 2018

www.computer.org

IEEE

IEEE COMPUTER SOCIETY


IEEE computer society
Looking for the BEST Tech Job for You?
Come to the **Computer Society Jobs Board** to meet the best employers in the industry—Apple, Google, Intel, NSA, Cisco, US Army Research, Oracle, Juniper...
Take advantage of the special resources for job seekers—job alerts, career advice, webinars, templates, and resumes viewed by top employers.
www.computer.org/jobs

COMPUTING edge

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

Printed with inks containing soy and/or vegetable oils

IEEE

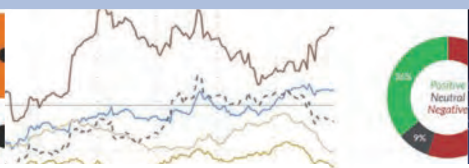
IEEE COMPUTER SOCIETY

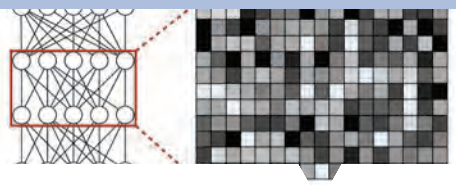
Circulation: ComputingEdge (ISSN 2469-7087) is published monthly by the IEEE Computer Society. IEEE Headquarters, Three Park Avenue, 17th Floor, New York, NY 10016-5997; IEEE Computer Society Publications Office, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720; voice +1 714 821 8380; fax +1 714 821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

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

OCTOBER 2018 • VOLUME 4, NUMBER 10

COMPUTING edge





40

Squeezing Deep Learning into Mobile and Embedded Devices

Magazine Roundup

Editor: Lori Cameron

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 design. Here are highlights from recent issues.

Computer

Mobile Decision Support and Data Provisioning for Low Back Pain

The authors of this article, from the August 2018 issue of *Computer*, present Back Pain Buddy, a mobile application offering decision support and coaching for people with low back pain (LBP). The application takes advantage of smartphones' powerful capabilities and provides a crowd-sourced decision support system for discovering treatments and a mobile sensing solution for collecting data about user activities that are crucial in LBP research.

Computing in Science & Engineering

Toward an Open, Sustainable National Advanced Computing Ecosystem

For over three decades, a largely organic process has brought into existence a somewhat disjointed and chaotic national research computing ecosystem that supports a collection of computational resources and services. Consequently, much of the national advanced research computing environment can be characterized by pockets of more coherent resources and services in a larger, less-coherent ecosystem. The rise in deployment

2469-7087/18/$33.00 © 2018 IEEE

of research computing resources on university campuses has added complexity to the situation. The National Science Foundation has made a foray into developing a more coherent environment via investments in XSEDE, the associated research computing resources, and critical common services and support for researchers. Read more in the September/October 2018 issue of *Computing in Science & Engineering*.

IEEE Annals of the History of Computing

Critical Failure: Computer-Aided Instruction and the Fantasy of Information

The history of the use of various kinds of computers in education involves frequent triumphalist claims about the inevitable automation of instruction and equally frequent declarations of the failure of this project. This article, which appears in the April–June 2018 issue of *IEEE Annals of the History of Computing*, situates both types of claims within a broader cultural understanding, one that holds that the human world is fundamentally informational and therefore amenable to improvement by computers.

IEEE Cloud Computing

Controlling User Access to Cloud-Connected Mobile Applications by Means of Biometrics

Cloud-connected mobile applications are becoming a popular solution for ubiquitous access to online services such as cloud data storage platforms. The adoption of such applications has brought security and privacy implications that are making individuals hesitant to migrate sensitive data to the cloud; thus, new secure authentication protocols are needed. In this article, which appears in the July/August 2018 issue of *IEEE Cloud Computing*, the authors propose a continuous authentication approach integrating physical (face) and behavioral (touch and hand movements) biometrics to control user access to cloud-based mobile services, going beyond one-off login.

IEEE Computer Graphics and Applications

LightPainter: Creating Long-Exposure Imagery from Videos

This article, which appears in the July/August 2018 issue of *IEEE Computer Graphics and Applications*, presents LightPainter, an interactive tool that promotes creative long-exposure photography through an intuitive drawing metaphor and flexible spatiotemporal mapping from videos to composite images. The authors discuss the power of software-defined exposure and the tool's capability of facilitating sophisticated long-exposure effects in challenging scenarios.

IEEE Intelligent Systems

Next-Generation Smart Environments: From System of Systems to Data Ecosystems

Digital transformation is driving a new wave of large-scale data-rich smart environments. The resulting data ecosystems present new challenges and opportunities in the design of intelligent systems and system of systems. Read more in the May/June 2018 issue of *IEEE Intelligent Systems*.

IEEE Internet Computing

OmniShare: Encrypted Cloud Storage for the Multi-Device Era

Two attractive features of cloud storage services are the automatic synchronization of files between multiple devices and the possibility of sharing files with other users. However, many users are concerned about the security and privacy of data stored in the cloud. Client-side encryption is an effective safeguard, but it requires all client devices to have the decryption key. Current solutions derive these keys from user-chosen passwords, which are easily guessed. In this article, which appears in the July/August 2018 issue of *IEEE Internet Computing*, the authors present OmniShare, the first scheme to combine strong client-side encryption with intuitive key distribution mechanisms to enable access from multiple client devices and sharing between users. OmniShare uses a novel combination of out-of-band channels (including QR codes and ultrasonic communication), as well as the cloud storage service itself, to authenticate new devices. The authors describe the design and implementation of OmniShare and explain how they evaluated its security, performance, and usability.

IEEE Micro

Perceived-Color Approximation Transforms for Programs that Draw

Human color perception acuity is not uniform across colors. This makes it possible to transform drawing programs to generate outputs whose colors are perceptually equivalent but numerically distinct. One benefit of such transformations is lower display power dissipation on organic light-emitting diode (OLED) displays. In this article, which appears in the July/August 2018 issue of *IEEE Micro*, the authors introduce Ishihara, a language for 2D drawing that lets programs specify perceptual-color equivalence classes to use in drawing operations, enabling compile-time and runtime transformations that trade perceived color accuracy for lower OLED display power dissipation.

IEEE MultiMedia

Toward Real-Time Delivery of Immersive Sports Content

Free viewpoint technology makes it possible to view video of sports content from any angle or position, but creating such content is currently a time-consuming process that can prevent real-time delivery. To address this problem, the authors of this article from the April–June 2018 issue of *IEEE MultiMedia* present an application framework that implements semi-automatic camera calibration, object extraction, object tracking, and object separation to seamlessly generate high-quality free viewpoint sports videos for handheld devices.

IEEE Pervasive Computing

Robotic Symbionts: Interweaving Human and Machine Actions

This article from the April–June 2018 issue of *IEEE Pervasive Computing* defines a category of human–robot interaction in which human and robotic actors work as a single unified system. The authors survey work from various fields including human augmentation systems such as extra fingers and arms, and other robots that operate in close proximity to the user. The discussed works highlight a close interplay between human and robotic actions where control decisions are made by both actors. Such a dyadic configuration can yield a synergistic outcome but requires that close attention be paid to the coordination between them. Using case studies from their own work, the authors discuss two main questions that must be addressed when designing such closely collaborative human–robot integrations: type of support and degree of control. The different choices that can be adopted for each of these design questions define a framework or classification that is useful for surveying existing and future research.

IEEE Security & Privacy

Botnet in the Browser: Understanding Threats Caused by Malicious Browser Extensions

Browser extension systems risk exposing APIs, which are too permissive and cohesive with the browser's internal structure, leaving a hole for malicious developers to exploit security-critical functionality. In this article, which appears in the July/August 2018 issue of *IEEE Security & Privacy*, the authors present a botnet framework based on malicious browser extensions and describe a range of attacks that can be launched in this framework.

IEEE Software

Code Reviewing in the Trenches: Challenges and Best Practices

Code review has been widely adopted by and adapted to open source and industrial projects. Code review practices have undergone extensive research, with most studies relying on trace data from tool reviews, sometimes augmented by surveys and interviews. Several recent industrial research studies, along with blog posts and white papers, have revealed additional insights on code reviewing "from the trenches." Unfortunately, the lessons learned about code reviewing are widely dispersed and poorly summarized by the existing literature. In particular, practitioners wishing to adopt or reflect on an existing or new code review process might have difficulty determining what challenges to expect and which best practices to adopt for their development context. Building on the existing literature, this article from the July/August 2018 issue of *IEEE Software* adds insights from a recent large-scale study of Microsoft developers to summarize the challenges that code-change authors and reviewers face, suggest best code-reviewing practices, and discuss tradeoffs that practitioners should consider.

IT Professional

Data Breaches: Public Sector Perspectives

A data breach exposes confidential, protected data to unauthorized access and manipulation. This study from the July/August 2018 issue of *IT Professional* examines the extent and occurrences of data breaches in the US public sector through an analysis of data breaches over a five-year period. This research is motivated by two questions: What are the current trends of data breaches in the public sector and how do contextual governmental factors impact these data breaches?

EDITOR'S NOTE

The Age of the Internet of Things

The Internet of Things (IoT) refers to the billions of connected devices that collect and share data—these devices can be as small as a lightbulb or as large as an airplane. Gartner estimates that around 8.4 billion IoT devices were in use in 2017, and that this number will likely reach 20.4 billion by 2020. The safety and security of these devices become more critical as they permeate our everyday lives.

Five articles in this issue of *ComputingEdge* focus on the IoT. In *IEEE Internet Computing*'s "A Principles-Based Approach to Govern the IoT Ecosystem," the authors propose the formulation of principles as a means to unify the multiple bodies and organizations involved in the governance of IoT devices and systems. In *Computer*'s "Taming the IoT: Operationalized Testing to Secure Connected Devices," the authors posit that end-to-end automation can normalize the IoT ecosystem and increase the velocity of system improvements and updates.

Cyber-physical systems (CPS) are mechanisms that are controlled or monitored by computer-based algorithms and are tightly integrated with the Internet, making them part of the IoT. Examples include smart grids and autonomous automobile systems. In *Computer*'s "The Rise of Intelligent Cyber-Physical Systems," the author expects that the CPS revolution will be more transformative than the IT revolution of the past four decades. The authors of *IT Professional*'s "How Do You Create an Internet of Things Workforce?" propose a new discipline that is focused on the IoT and CPS, with the goal of educating and training future workers in this growing field.

Wearables are also IoT devices, as they contain sensors that collect and transmit data from the wearer to the Internet. In *IEEE Pervasive Computing*'s "Squeezing Deep Learning into Mobile and Embedded Devices," the authors describe phone, watch, and embedded prototypes that can locally run large-scale deep networks that process audio, images, and inertial sensor data.

Other topics in this issue include social media and education. *Computing in Science and Engineering*'s "Did Everybody Come?" evaluates author Clay Shirky's popular claim that social networking has the potential to change society by making it easier for people to come together. *IEEE Intelligent Systems*' "Aspect-Based Extraction and Analysis of Affective Knowledge from Social Media Streams" combines factual and affective knowledge extracted from rich public knowledge bases to analyze emotions expressed toward specific targets in social media.

The history of the use of various kinds of computers in education involves frequent claims about the inevitable automation of instruction and equally frequent declarations of the failure of this project. *IEEE Annals of the History of Computing*'s "Critical Failure: Computer-Aided Instruction and the Fantasy of Information" situates both claims within a broader cultural understanding that holds that the world is fundamentally amenable to improvement by computers. Finally, in *IEEE Software*'s "How Best to Teach Global Software Engineering?," pioneering educators discuss how they inject realism into global software engineering education.

 2469-7087/18/$33.00

IEEE computer society

PURPOSE: The IEEE Computer Society is the world's largest association of computing professionals and is the leading provider of technical information in the field.
MEMBERSHIP: Members receive the monthly magazine *Computer*, discounts, and opportunities to serve (all activities are led by volunteer members). Membership is open to all IEEE members, affiliate society members, and others interested in the computer field.
COMPUTER SOCIETY WEBSITE: www.computer.org
OMBUDSMAN: Direct unresolved complaints to ombudsman@computer.org.
CHAPTERS: Regular and student chapters worldwide provide the opportunity to interact with colleagues, hear technical experts, and serve the local professional community.
AVAILABLE INFORMATION: To check membership status, report an address change, or obtain more information on any of the following, email Customer Service at help@computer.org or call +1 714 821 8380 (international) or our toll-free number, +1 800 272 6657 (US):

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

PUBLICATIONS AND ACTIVITIES
***Computer*:** The flagship publication of the IEEE Computer Society, *Computer*, publishes peer-reviewed technical content that covers all aspects of computer science, computer engineering, technology, and applications.
Periodicals: The society publishes 13 magazines, 19 transactions, and one letters. Refer to membership application or request information as noted above.
Conference Proceedings & Books: Conference Publishing Services publishes more than 275 titles every year.
Standards Working Groups: More than 150 groups produce IEEE standards used throughout the world.
Technical Committees: TCs provide professional interaction in more than 30 technical areas and directly influence computer engineering conferences and publications.
Conferences/Education: The society holds about 200 conferences each year and sponsors many educational activities, including computing science accreditation.
Certifications: The society offers two software developer credentials. For more information, visit www.computer.org/certification.

IEEE

revised 03 July 2018

A Principles-Based Approach to Govern the IoT Ecosystem

Virgilio A.F. Almeida and Benjamin Goh • *Harvard University*

Danilo Doneda • *Rio de Janeiro State University*

The difference between a good and bad Internet of Things depends on society's ability to construct effective IoT governance models. This article proposes the formulation of principles as a means to unify the multiple bodies and organizations involved in the IoT governance ecosystem.

With the attack on Dyn in 2016, the Internet of Things' security and its potential impact on the Internet are once again in the spotlight.[1] The Dyn attack, aimed at the Internet's domain name server (DNS) infrastructure, disrupted multiple major service providers, including Twitter, Netflix, Spotify, Airbnb, Reddit, and *The New York Times*.[2] In a public announcement in September 2015, the FBI warned about the use of IoT devices and the potential virtual and physical threats they might pose.[3] As Vint Cerf emphasized,[4] the difference between good and bad IoT depends on society's ability to construct effective IoT governance models. In this article, we discuss ideas for the development of the IoT governance ecosystem.

IoT's logic arises primarily from tech companies, who wish to use increased connectivity to market products that provide greater convenience and more personalized services. Amazon's Alexa, Google's driverless car, and the Fitbit Flex are all products that ride this new wave of digital convenience. However, beyond the consumer level, IoT applications are increasingly used in industries, such as energy management systems, industrial automation, and in management of urban facilities, such as smart grids and smart traffic lights. Used in this way, IoT poses serious cybersecurity issues, creating "new risks in complex ecosystems."[5] Such IoT systems create new risks around privacy and security protections, especially when they're used in mission-critical systems. In essence, IoT applications amplify vulnerabilities in existing software and hardware.

To ensure safety, security, and privacy in the IoT ecosystem, governments, civil society, the private sector, and academia must be at the table to discuss new governance mechanisms that minimize the risks introduced by IoT. The consequences of delaying the construction of rules, norms, and regulations for IoT are potentially catastrophic.

Minimizing or Mitigating IoT Security Threats

There's no doubt that IoT services provide services and efficiency that can improve welfare. However, it opens up new levels of vulnerabilities that raises further governance questions: for example, while ISPs used to be the only one able to retrieve web browsing history from someone's personal WiFi, the explosion of devices connected through the home can now reasonably predict a person's activities at home, raising new privacy and security concerns. Security researchers at Princeton University found that "the contents, patterns, and metadata of network traffic can all reveal sensitive information about a user's online activity."[6] In particular, they found that even with encrypted traffic, a network observer can use network send/receive rates to tell if a user is sleeping, or if there's a change in frequency of motion to determine if the house is occupied or if guests are coming.[6]

The US Federal Trade Commission's (FTC's) 2015 staff report on IoT classifies security threats in the following three ways:

- enabling unauthorized access and misuse of personal information;
- facilitating attacks on other systems; and
- creating safety risks.[7]

First, the risk to unauthorized access and misuse of personal information isn't new, especially if you consider the vulnerability of social media accounts from being compromised, or that a USB inherently has insecure design flaws.[8] However, IoT creates a new urgency to the problem, because an individual's security is only as strong as the weakest link, and an IoT system might create more opportunities for "lateral movement" to compromise someone's security. Whereas it might be convenient for us to rely on our phones for all IoT services (such as thermostats), malware that infects smartphones might compromise our safety at a scale larger than before.[9]

Second, IoT devices are, by definition, devices that have the ability to connect to the Internet. Thus, vulnerabilities in any IoT device have the potential to become an attack vector through which a malicious actor causes harm to others. The most popular form of such vulnerabilities come in the form of the Mirai botnet, which takes advantage of industry negligence toward IoT devices to compromise devices for nefarious use.[1]

Third, new dependencies are created by IoT services, which can create new sources of risk to human safety. The US Food and Drug Administration (FDA) found that pacemakers and defibrillators by St. Jude Medical contain cybersecurity risks that make them highly vulnerable to attack, potentially affecting the lives of tens of thousands of patients with cardiac devices.[10] Security researchers have also famously shown how they could remotely compromise the Chrysler Jeep Cherokee's entertainment system, rewrite its firmware, and control the car by sending commands to critical systems (such as the brakes).[11]

Security and privacy protection are key for a "good" IoT. However, IoT applications create different types of privacy risks. Smart TVs, for example, through beaconing technology and cross-device tracking, allow all home devices to share information without our knowledge. Along these lines, Amazon recently agreed to hand federal courts data gathered from an Echo speaker to assist in investigations in a murder case.[12] It isn't clear how much information the Amazon Echo collects, but it also raises important questions about privacy — is the home still "private," or does one forgo privacy protections by purchasing an Echo? Finally, distinct from privacy risk is the IoT's potential in creating surveillance risks. Whereas privacy is most famously defined by Louis Brandeis to be the "right to be left alone," surveillance risks occur when the government has an abundance of tools to monitor individual behavior. In the Berkman Klein Center's *Don't Panic* report, for example, the authors found that metadata is unlikely to become encrypted, which provides government officials a wealth of data such as "location data from cell phones and other devices, telephone calling records, and header information in e-mail" that can fuel surveillance.[13]

The IoT Governance Ecosystem

The IoT governance ecosystem has many players with very different legal statuses. They operate on many different layers on municipal, national, and international levels, driven by technical innovation, user needs, market opportunities, and political interests. The specific form of each component of the ecosystem must be designed according to the very specific needs and nature of the individual issue. There's no "one size fits all" solution for IoT governance.

Many agencies and organizations deal with guidelines and regulation of IoT devices. On the municipal level, the City of New York[14] has proposed a common framework to help agencies develop policies for IoT with the following goals:

- provide a common framework to help governments develop and expand policies and procedures related to the IoT;
- ensure openness and transparency regarding the use of public space or assets for smart city technologies; and
- advance the public dialogue about how government, the private sector, and academia can collaborate to ensure these technologies are used in a way that maximizes public benefit.[15]

An example of a municipal rule for IoT is "All IoT devices and network equipment installed on city property should have clear site license agreements and established terms of service governing who is responsible for ongoing operations, maintenance, and the secure disposal of equipment."

On the national level, many countries have initiatives to create regulations and standards for IoT applications. In the US, several agencies — including the Food and Drug Administration, the Federal Communications Commission, the FTC, and the National Highway Traffic Security Administration — are reviewing some aspects of IoT.[16] As the technology moves into healthcare, and data from wearable health devices flows more from consumers' wrists to companies, the Food and Drug Administration (FDA) is keeping interest in the evolution of IoT applications. The US Department of Energy (DOE) established the Federal Smart Grid Task Force, with experts from 11 different federal agencies to coordinate strategies to promote integration of smart-grid technologies and practices. At the international level, different organizations have proposed guidelines and standards for the IoT. The Internet Society (ISOC),[17] the IETF,[18] and the International Telecommunication Union (ITU)[19] have published reports and recommendations of technical standards to enable IoT on a global scale.

IoT Governance Principles

So, out of this a natural question arises: What could be used to "glue" different groups and interests together in a global IoT governance ecosystem? Even considering the importance of IoT governance, the way it can be structured is absolutely open for debate. Nonetheless, the vectors this structure shall follow can be drawn from the reflection utterly made in the face of the development of governance tools to act on the Internet environment. Common principles could be the element that will put together different interests in an environment in an inclusive, effective, and legitimate governance framework. They could contribute to contextualizing the IoT as part of global resources that should be managed in the public interest. In this sense, we chose a set of applicable principles developed in the NETmundial Multistakeholder Conference.[20]

Governments and several stakeholder groups, including civil society, private sector, and academia, gathered to discuss issues and principles for Internet governance and roadmap actions for the Internet's future evolution. Among the issues discussed, the scope of Internet governance was preeminent, in the sense of the tension between those who see Internet governance as a mostly technical matter (with, for example, IP numbers, routing and specifications, DNS, and critical resources) and others who approach Internet governance as something that must comprehend and factor important social and political issues, such as privacy, freedom of expression, and human rights in a general sense.

The final result, the NETmundial Declaration, encompassed principles both of a technical nature as well as non-technical ones. Some of these principles can be deemed as guidance to IoT governance, as a relevant part of IoT's impact can be related to them. For example, one principle refers to the structure of the Internet governance ecosystem, which should be built on democratic, multistakeholder processes, ensuring the meaningful and accountable participation of all stakeholders, including governments, the private sector, civil society, the technical community, the academic community, and users. This principle reiterates the importance of having civil society representatives in governance bodies. In the case of IoT, this should be a key principle, in particular because of the massive presence of IoT devices on the consumer side. Two other principles could be used in the construction of the global IoT governance ecosystem: first, governance models should be open, participative, transparent, and consensus-driven; and second, Internet governance should be carried out through a distributed, decentralized, and multistakeholder ecosystem.

Issues related to security and privacy rise to the fore as IoT's influence permeates our daily lives. Such issues then reflect onto the NETmundial principle about privacy that states, "The right to privacy must be protected. This includes not being subject to arbitrary or unlawful surveillance, collection, treatment and use of personal data" (www.netmundial.org/principles). This principle encompasses data protection as well. In fact, to the extent that IoT provides for a vast number of devices to be connected to the Internet, it happens that several of them gather personal data. Many of them are strictly sensors that are responsive to personal activities. This makes for a concrete increase in the volume of personal data gathered. It also makes the case regarding what these devices can do: they collect far more personal data than is reasonably expected, deemed fair, or authorized, and they proceed to the treatment of the personal data they collect with low security. These two points are linked to another characteristic tendency of IoT, which is the proliferation of small and simple devices, in general too simple and cheap to include safeguards about excessive and unfair collection of personal data or to implement data security at a reasonable level.

Eventually, these IoT weaknesses can be addressed through a conjunction of other principles present in the NETmundial declaration, particularly the principles of accountability and transparency. And, moreover, there's the fact that the IoT per se exponentially expands the number of devices attached to the Internet (often small and cheap ones) and these devices, if expected to comply with privacy and security rules, shall be submitted to technical standards that emphasize this compliance. In this sense, some of the Internet governance principles of the NETmundial Declaration can be used as a basis to assemble interested stakeholders in an open and participative dialogue for constructing the IoT governance ecosystem.

References

1. N. Woolf, "DDoS Attack That Disrupted Internet Was Largest of Its Kind in History, Experts Say," *The Guardian*, 26 Oct. 2016; www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet.

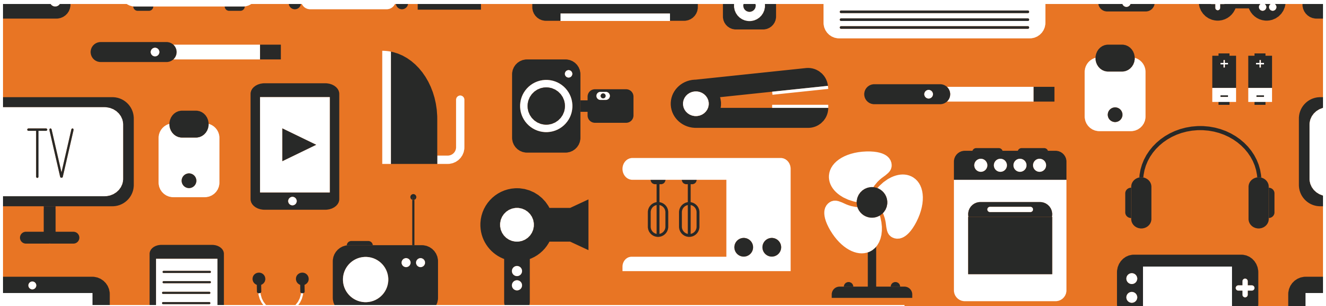
This article originally appeared in
IEEE Internet Computing, *vol. 21, no. 4, 2017.*

2. N. Perlroth, "Hackers Used New Weapons to Disrupt Major Websites Across U.S.," *The New York Times*, 21 Oct. 2016; www.nytimes.com/2016/10/22/business/internet-problems-attack.html
3. US Federal Bureau of Investigation, *Internet of Things Poses Opportunities for CyberCrime*, Alert Number I-091015-PSA, 10 Sept. 2015; www.ic3.gov/media/2015/150910.aspx.
4. F. Berman and V.G. Cerf, "Social and Ethical Behavior in the Internet of Things," *Comm. ACM*, vol. 60, no. 2, 2017, pp. 6–7.
5. European Commission's High Level Group of Scientific Advisors, *Cybersecurity in the European Digital Single Market*, scientific opinion no. 02, Scientific Advice Mechanism, European Union, 24 Mar. 2017; https://ec.europa.eu/research/sam/pdf/sam_cybersecurity_report.pdf.
6. N. Apthorpe, D. Reisman, and N. Feamster, "A Smart Home Is No Castle: Privacy Vulnerabilities of Encrypted IoT Traffic," US Federal Trade Commission (FTC), 2016; www.ftc.gov/system/files/documents/public_comments/2016/10/00022-131586.pdf.
7. FTC, *Internet of Things: Privacy & Security in a Connected World*, FTC staff report, 2015; www.ftc.gov/system/files/documents/reports/federal-trade-commission-staff-report-november-2013-workshop-entitled-internet-things-privacy/150127iotrpt.pdf.
8. J. Condliffe, Jamie. "USB Has a Fundamental Security Flaw That You Can't Detect." *Gizmodo*, 31 July 2014; http://gizmodo.com/usb-has-a-fundamental-security-flaw-that-you-cant-detec-1613833339.
9. R. Brandon, "App-Installing Malware Found in over 1 Million Android Phones," *The Verge*, 30 Nov. 2016; www.theverge.com/2016/11/30/13792846/googlian-android-malware-install-app-security.
10. US Food and Drug Administration, "Cybersecurity Vulnerabilities Identified in St. Jude Medical's Implantable Cardiac Devices and Merlin FDA Safety Comm.," FDA safety comm., 9 Jan. 2017; www.fda.gov/MedicalDevices/Safety/AlertsandNotices/ucm535843.htm.
11. A.C. Estes, "Hackers Have the Power to Remotely Hijack Half a Million Chrysler Cars," *Gizmodo*, 21 July 2015; http://gizmodo.com/hackers-have-the-power-to-remotely-hijack-half-a-millio-1719233440.
12. I. Smith, "Amazon Releases Echo Data in Murder Case, Dropping First Amendment Argument," *PBS News Hour*, 8 Mar. 2017; www.pbs.org/newshour/rundown/amazon-releases-echo-data-murder-case-dropping-first-amendment-argument/.
13. U. Gasser et al., *Don't Panic: Making Progress on the "Going Dark" Debate*, tech. report, Berkman Center for Internet & Soc., Harvard Univ., 2016; https://cyber.harvard.edu/pubrelease/dont-panic/Dont_Panic_Making_Progress_on_Going_Dark_Debate.pdf.
14. The New York City Mayor's Office of Technology and Innovation, *NYC Guidelines for the Internet of Things*, Sept. 2016; www.nyc.gov/iot.
15. D. Castro and J. New, *Everything the U.S. Government Is Doing to Help the Private Sector Build the Internet of Things*, tech. report, Center for Data Innovation, Information Technology & Innovation Foundation, 12 Dec. 2016.
16. M. Ravindranath, "Who's in Charge of Regulating the Internet of Things?" *NextGov*, 1 Sept. 2016; www.nextgov.com/emerging-tech/2016/09/internet-things-regulating-charge/131208/.
17. *The Internet of Things (IoT): An Overview - Understanding the Issues and Challenges of a More Connected World*, white paper, The Internet Society, 2015; www.internetsociety.org/doc/iot-overview.
18. A. Keränen and C. Bormann, "Internet of Things: Standards and Guidance from the IETF," *IETF J.*, Apr. 2016; www.internetsociety.org/publications/ietf-journal-april-2016/internet-things-standards-and-guidance-ietf.
19. V. Almeida, "The Evolution of Internet Governance: Lessons Learned from NETmundial," *IEEE Internet Computing*, vol. 18, no. 5, 2014, pp. 65–69.
20. W. Kleinwächter and V. Almeida, "The Internet Governance Ecosystem and the Rainforest," *IEEE Internet Computing*, vol. 19, no. 2, 2015, pp. 64–67.

Virgilio A.F. Almeida is a faculty associate at the Berkman Klein Center for Internet and Society at Harvard University, and a professor in the Computer Science Department at the Federal University of Minas Gerais (UFMG), Brazil. His research interests include cyber policies, large-scale distributed systems, the Internet, and social computing. Almeida has a PhD in computer science from Vanderbilt University. Contact him at virgilio@dcc.ufmg.br or valmeida@cyber.law.harvard.edu.

Benjamin Goh is a master's in public policy candidate at the John F. Kennedy School of Government, Harvard University. His research interests surround cybersecurity strategy, and the appropriate role of government in the digital world. Goh graduated with BS degrees, summa cum laude, with double honors in economics and international relations from New York University. He was named the Ellie and David Werber Research Scholar in Social Sciences at NYU, and received the Fiona McGillivray Prize for his thesis on the political economy of Internet surveillance. Contact him at Benjamin_Goh@hks17.harvard.edu.

Danilo Doneda is a professor of civil law at the Law School of the Rio de Janeiro State University (UERJ). His research interests include private law and regulation, privacy, and data protection. Doneda has a PhD in civil law from UERJ. Contact him at danilo@doneda.net.



Taming the IoT: Operationalized Testing to Secure Connected Devices

Trevor Pering, Kathy Farrington, and Thorsten Dahm, Google

Operationalized testing of built-environment IoT infrastructure is a key aspect of fostering security and development best practices. End-to-end automation can normalize the ecosystem and increase the velocity of system improvements and updates.

FROM THE EDITOR

A testing environment has become an essential tool for validating large software systems. Understanding how small changes can affect a complex system is often not possible because the scale and abstractness make it hard for a developer to comprehend its consequences. In addition, the system was probably built by many people over time, and no one person has all the knowledge. Building-scale IoT infrastructure is no different. In this article, the authors describe how a test-driven environment can stem many of the operational problems the IoT is experiencing today. —*Roy Want*

Security is a key factor in the success of enterprise Internet of Things (IoT) systems that can be enhanced by operationalizing test-driven development (TDD) through a process called device automated qualification (DAQ). In complex software systems, security flaws are nearly unavoidable and always pressing when devices are connected to the Internet, and the speed at which they can be addressed is a key factor in their mitigation. DAQ aims to normalize an ecosystem of IoT devices by applying TDD at multiple stages, reducing the complexity of applying security policies and increasing the velocity by which policies and principles can be enforced. This is one path to avoiding IoT's "certification

Published by the IEEE Computer Society 2469-7087/18/$33.00

EDITOR ROY WANT
Google; roywant@google.com

quagmire," as discussed in this column's last installment,[1] by focusing on action (active tests) rather than policy (stated requirements).

Many IoT systems suffer from slow update cycles because they were initially intended to be single-use devices or have been closely tied to heavily regulated physical systems, which also change slowly. TDD increases the velocity of IoT software development, leading to faster security updates by making it easier for device developers to update their code; furthermore, it provides a mechanism for the security requirements (as tests) to be pushed into the device ecosystem itself.

A *built environment* is a physical space, such as a building, that is explicitly built with a specific purpose in mind. In the architectural study *How Buildings Learn: What Happens After They're Built*,[2] author Stewart Brand outlines several different layers of a typical building along with how often each layer changes. The outermost skin of a building, for example, changes on average every 20 years (see Figure 1). A building's services would be heating or lighting, which can change as infrequently as every 7 to 15 years. As buildings are brought online, these physical constructs form the foundation for connected "things."

The Internet, which has no firm roots in the physical world, changes constantly. IoT for a built environment—the quintessential smart building—is therefore sitting on a mismatch in update cycles that easily span several orders of magnitude. Addressing security in such an environment, therefore, involves being able to rapidly patch systems that have traditionally been left in situ for years without any updates.

From a security perspective, it is difficult to trust IoT devices.[3] There are simply too many of them, and because of their constrained nature there are often compromises that weaken security overall. Most IoT devices are typically focused on a physical task rather than being general-purpose computing platforms. Accordingly, the core principle of "trust but verify" applies. Assuming that every device is doing its best to be secure, it is still important to externalize and automate the process to identify and fix problems before they can cause harm.

DEVELOPMENT AND OPERATIONS

The solution to the security problem is not just the obvious "quickly push a security patch to a device" method (which can reasonably be addressed with improved firmware update capabilities). The greater challenge lurks in the process of pushing requirements into the building device development ecosystem. It's not enough to simply say what must be done, because until a device correctly implements (or removes) a feature, the vulnerability still exists. The function of these devices primarily represents a physical system, and are often embedded controllers designed by control engineers whose primary considerations are far removed from the security of the device itself.

For many years, protocols with weak or nonexistent authentication, encryption, or integrity features have been regularly used by IoT systems. Consider the popular telnet protocol, which is a long-standing standard for accessing networked devices. Unfortunately, it opens the door to compromise because it uses unencrypted and poorly authenticated communication. The most straightforward solution to this is to simply disallow it, requiring the use of alternate protocols such as SSH (Secure Shell). In a perfect world, this would be enough; however, the question is not just what the device is designed to do, but rather what it actually does.

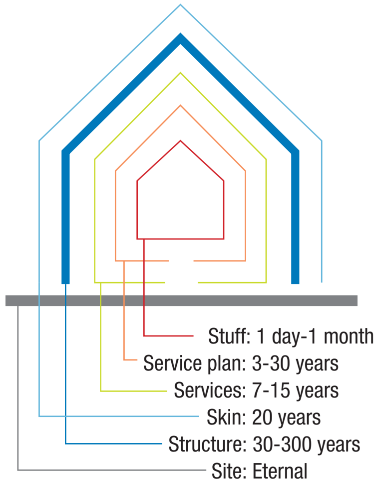


Figure 1. A building's internal lighting services change on the order of every 7 to 15 years, much more rapidly than the building structure itself.

There are any number of failure points that might remain without proper enforcement, including the following scenarios:

- a product owner doesn't know they should disable a protocol;
- a developer doesn't remove all of the offending code (just some uses of it);
- the documentation doesn't mention the protocol, even though the device implements it;
- people don't always read the documentation, even if it's there;
- a system might be misconfigured, enabling a protocol even though it should be disabled; and
- the replacement protocol is misconfigured, as it's new and not well understood.

For an advanced smart building, there is a long device pipeline starting with manufacturers and extending to

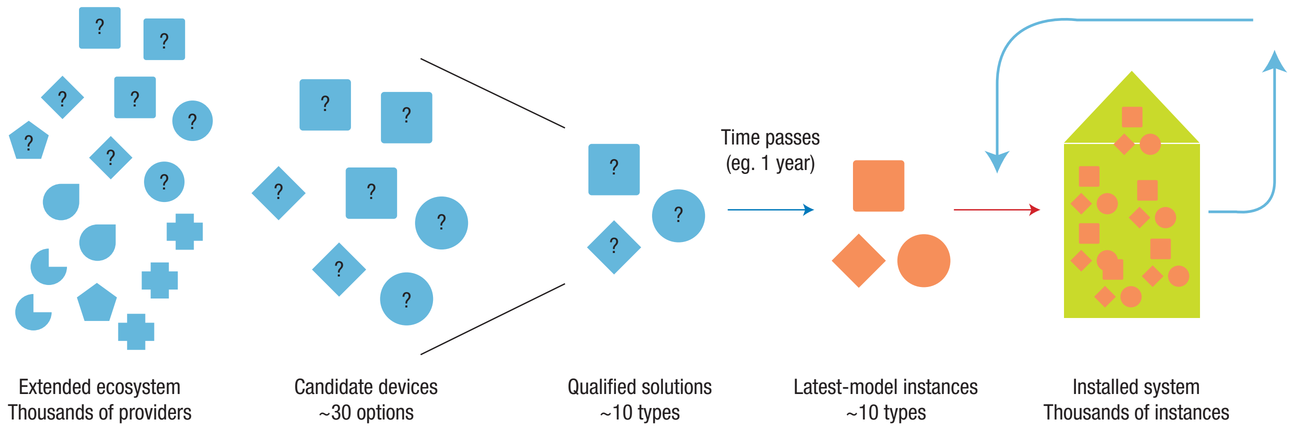


Figure 2. Operationalized testing for the built environment spans all the way from an extensive ecosystem of device manufacturers to a live building.

installed systems (see Figure 2). Initially, the pool of potential IoT devices for built environments numbers in the thousands, many of which are regional or not specific in their function. In the best case, the initial design will specify two to three candidates for each particular function in a building. Individual candidates for each type will then be selected, which is on the order of 10 qualified device types per building. Once the devices are selected, their integration into the building itself is architected, which could take a year or more. During this time, some devices themselves evolve (updated firmware, hardware specs, and so on), requiring an additional latest-model test phase. During the construction process, thousands of actual devices will be installed, and a continual process of monitoring and maintenance supports ongoing building operation.

One mechanism currently employed to address qualifying devices is manual qualification, where devices are considered and ultimately placed into a new built environment. Device manufactures respond to a tender request from a building's master systems integrator (MSI), who then qualifies devices to make sure they meet required performance and security specifications. Before devices are installed in a building, they are again tested to make sure they continue to conform to requirements (which could have evolved in the two to three years it can take to design and construct a building). Finally, after installation, the entire system is tested in situ to verify functionality (and security through requirements).

This model only works if the rate of change is low, essentially matching the physical progression of the device itself. What happens when updates are more frequent, necessitated by more stringent security requirements? The manual process simply does not scale, and either becomes a roadblock to constructing smart buildings (because IT organizations won't allow it), or breaks down by allowing vulnerabilities to creep inside the building (which isn't very smart).

DAQ

The DAQ framework (https://github.com/faucetsdn/daq) is a software tool designed to apply security principles to IP-based devices throughout their lifecycle: before, during, and after construction. As an open source project, it's available to all device developers to help ensure that their products meet requisite security guidelines. Integrators can utilize the system to more comprehensively assess the suitability of a wider selection of devices, and it can even be applied in situ to an (already) built environment to ensure the ongoing conformance of systems. So, instead of mandating that devices "shall not use telnet," automated testing can ensure that policy is enforced at all stages of the design and during operation (automation enables continuous testing).

The DAQ system architecture (see Figure 3) borrows heavily from Faucet SDN,[4] which is used to validate network switches in an enterprise environment—a suite of integration tests runs against target devices that implement the OpenFlow networking standard. One notorious problem with standards is that they are ambiguous or intentionally vague: words don't always translate into the same behavior when interpreted by different developers. By codifying the required behavior in a test, the exact intent that an organization requires can be unambiguously specified, decreasing the overall time to conformance for a particular feature or behavior. (It might not even be conformance to the entire standard, just applying it to the parts that matter.)

The primary difference between Faucet and DAQ lies in the nature of the device under test. First, network switches have a well-defined set of behaviors (specified in the OpenFlow standard), which is expected across all devices. Second, switching functionality is inherently testable because packets injected into the system result in observable packets out as an indication of success or failure. Third, the network switches occupy a trusted place in a building's infrastructure.

None of these properties hold true for a heterogeneous collection of embedded IoT computing devices interacting with physical systems. Instead, DAQ tests a wide variety of common attributes (such as exposed telnet ports) that can be generally applied to all devices connected on the network, rather than their intended functionality (for example, does a light switch actually turn on lights?). Scalability becomes paramount, not only in the ability to reach a large number of device manufacturers, but also in the community's ability to write new tests to address potential vulnerabilities.

The DAQ and Faucet qualification model is an externalized form of the standard TDD development methodology. With TDD, a developer would typically write tests to drive the specific behavior of the code, testing that it conforms to internal expectations. With DAQ, the focus is on compliance to external criteria. In both cases, however, the general model of "write a test to reify the desired behavior" exists: use the test to drive and validate development.

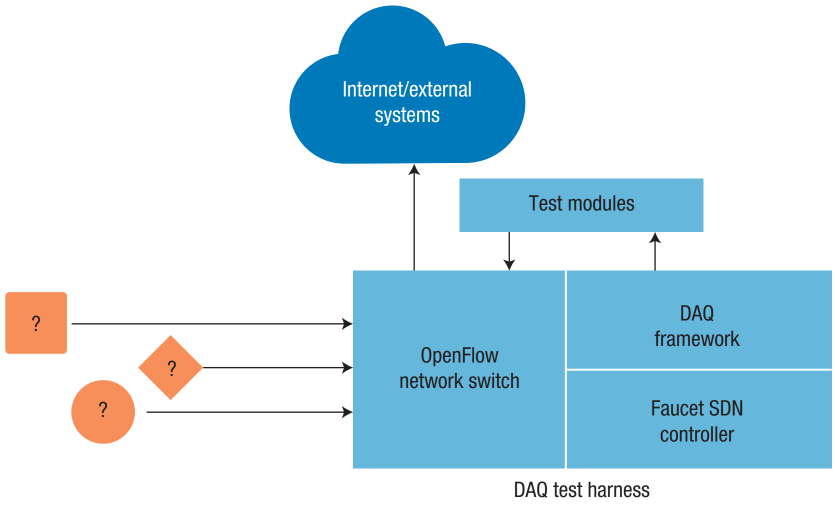


Figure 3. The device automated qualification (DAQ) system architecture uses dynamic network switching to test devices in both lab and live environments.

TEST COVERAGE

The range of tests required for security spans a number of different layers of a device's stack, from vulnerabilities due to buffer overflows to the validation of Transport Layer Security (TLS) certificate chains. Furthermore, once in place, the qualification process can be used to normalize devices' behavior in terms of communication with upper layers of the system, such as the format of telemetry ingested into an upstream database. Normalization enables anomaly detection by making patterns in an otherwise uniform dataset easier to analyze, increasing the system's ability to detect subtle forms of intrusion or misbehavior.

As requirements for devices expand, the test suite can expand along with it, encouraging ecosystem compliance much in the same way as a browser's HTML5 compliance score. Note that the tests involved are not new: DAQ provides a new mechanism by which to apply them. A sampling of specific tests that merit automation include (but are not limited to):[5]

- low-level networking: port scanning, checksum validation, and buffer overflows;
- core networking services: Dynamic Host Configuration Protocol (DHCP), Domain Name System (DNS), and Network Time Protocol (NTP);
- service vulnerabilities: default passwords and unencrypted communication protocols;
- system architecture: weak authentication schemas or bad access control;
- communication patterns: unexpected outgoing connections and network broadcast;
- encryption fundamentals: entropy monitoring, TLS validation, 802.1x (IEEE standard for port-based network access control), and key rotation;
- device management: automated firmware updates and device health reporting; and
- data telemetry: schema conformance and standard encodings.

At the basic levels of the stack (in other words, starting with low-level networking tests), expectations are clear: conform to baseline standards and limit vulnerabilities. Testing conformance for device management and data telemetry, however, moves into a territory that is much less standardized across a fleet of diverse devices. Firmware updates in particular are problematic because there is currently no standard way for a building to manage updates across all its devices, which come from different manufacturers. This directly leads to security vulnerabilities because it becomes much more onerous to address pressing security problems. The need for additional standardization in this space is clear.

OPERATIONALIZED QUALIFICATION

Operationalization is the process of putting something into continuous use: it's not enough for it to be available, used, or even useful. Because of the physical nature of construction projects, problems of scale are often solved by throwing more people or time at the project. However, good software engineering applies streamlined operations and robustness to maximize scalability.

To scale the built IoT ecosystem, where devices need to continually be monitored and updated, the entire "building" process should take a page or two from the *Site Reliability Engineering* playbook[6] that has grown out of the (relatively) recent expansion of cloud computing. If our buildings are to be trusted, managing them must be a seamless and error-free process. Not only does this mean increasing the confidence of deployments or updates with a "push on green" mentality, but it requires that problems can be isolated to failure domains that are easily identifiable. Established best practices such as a carefully controlled rollouts of any changes can now be applied. Comprehensive automated testing is a means to this end.

One key aspect of operationalization is ease of use; additionally, the system must be open (allowing everybody and anybody to use it), robust, and reliable. It needs to enable testing en mass (such as a room filled with a hundred devices to test). Many people using it will not have much experience with computing systems, as they are experts in other fields such as mechanical or environmental engineering. This is a tall order for any system, but the need is clear and the fundamentals are sound.

IoT in the built environment is a complex opportunity with potentially great rewards, but also significant risk. Only by directly addressing this risk can the industry evolve the concept of a smart building into one that rivals our mobile devices and cloud presence. Operationalized testing with DAQ is just one way to bridge the gap between the building and IT industries, collaboratively working together to tame the World Wide Wild of the IoT. ■

REFERENCES

1. J. Voas and P.A. Laplante, "IoT's Certification Quagmire," *Computer*, vol. 51, no. 4, 2018, pp. 86-89.
2. S. Brand, *How Buildings Learn: What Happens After They're Built*, Penguin Books, 1995.
3. T. Yu et al., "Handling a Trillion (Unfixable) Flaws on a Billion Devices: Rethinking Network Security for the Internet-of-Things," *Proc. 14th ACM Workshop Hot Topics in Networks* (HotNets-XIV), article no. 5, 2015; doi:10.1145/2834050.2834095.
4. J. Bailey and S. Stuart, "Faucet: Deploying SDN in the Enterprise using OpenFlow and DevOps for Rapid Development," ACM Queue, vol. 14, no. 5, 2016; doi:10.1145/3012426.3015763.
5. *Application Security Requirements for IoT Devices*, Google, 2017; https://partner-security.withgoogle.com/docs/iot_requirements.
6. *Site Reliability Engineering*, B. Beyer et al., eds., O'Reilly, 2016.

TREVOR PERING is a senior software engineer at Google. Contact him at peringknife@google.com.

KATHY FARRINGTON is a technical program manager at Google. Contact her at kathfar@google.com.

THORSTEN DAHM is a network security engineer at Google. Contact him at thorstendlux@google.com.

This article originally appeared in Computer, *vol. 51, no. 6, 2018.*

IEEE COMPUTER SOCIETY: Be at the Center of It All

IEEE Computer Society membership puts you at the heart of the technology profession—and helps you grow with it.

Here are 10 reasons why you need to belong.

1. **Training** that sharpens your edge in Cisco, IT security, MS Enterprise, Oracle and more.
2. **Certifications** and exam preparation that set you apart.
3. **Industry intelligence,** including *Computer,* myCS, Computing Now, and myComputer.
4. Over 200 annual **conferences** and technical events held worldwide.
5. 300-plus chapters and more than 30 technical committees **keep you connected.**
6. **Access** to hundreds of books, 13 technical magazines and 20 research journals.
7. **Deep discounts** on magazines, journals, conferences, symposia and workshops.
8. Opportunities to **get involved** through speaking, publishing and volunteering opportunities.
9. **Scholarships awarded** to computer science and engineering student members each year.
10. **A robust jobs board,** plus videos, articles and presentations to help you land that next opportunity.

IEEE Computer Society—keeping you ahead of the game. Get involved today.

www.computer.org/membership

IEEE computer society

Did Everybody Come?

Charles Day
American Institute of Physics

Clay Shirky's influential book *Here Comes Everybody: The Power of Organizing Without Organizations* (Penguin Books, 2008) explored the potential for Internet-based social networking to change society by making it easier for people to communicate. In particular, Shirky envisioned "mass amateurization":

> *Our social tools remove older obstacles to public expression, and thus remove the bottlenecks that characterized mass media. The result is the mass amateurization of efforts previously reserved for media professionals.*

When Shirky wrote the book, only 5 percent of US households had a smartphone. Geolocation, which makes ride-sharing apps like Uber possible, was in its infancy. Most Netflix customers opted to receive movies via mailed envelopes rather than the company's new streaming service.

Still, many of the current features of our digital lives were already in place 10 years ago. Facebook opened to the general public in 2006. By 2008, it had 100 million users. Twitter and MySpace were also on the scene. Online magazine *Slate* was 12 years old. Personal blogging was flourishing.

So, to what extent *did* everybody come? Perhaps the biggest validation of the nascent trends that Shirky spotted was the Women's March on Washington, DC. The day after Donald Trump was elected US President, Teresa Shook, a retiree living on the Hawaiian island of Maui, took to Facebook to urge her friends to march on the nation's capital in protest. Other individuals made the same plea on social media. And so on 20 January 2017, half a million people joined Shook in Washington, and an estimated three million marched in similar events around the world.

> In my view, Shirky overestimated how much people, campaigning aside, really do want to come together.

On the other hand, it's harder to claim that mass amateurization of journalism and other media is truly with us. Granted, YouTube has made stars of the likes of Joseph Garrett, who, in the persona of an orange cartoon cat called Stampylonghead, posts daily Minecraft videos for his six million subscribers. But for every Stampylonghead, there are legions of YouTubers like Anti, whose physics-themed Minecraft videos garner a few hundred views. The ease with which content can be created, discovered, and shared has increased the premium on quality. Garrett is successful because he's talented enough to stand out amid the 400 hours of content that gets uploaded to YouTube every minute.

The mainstreaming of blogging also seems to contradict mass amateurization. What Nate Silver and Ezra Klein—two prominent bloggers—publish now looks just like online journalism.

A final trend that contradicts *Here Comes Everybody* is how digital technology is making people *less* social. Thanks to Internet-mediated home delivery, staying at home is more attractive.

 Published by the IEEE Computer Society

Cloud-based streaming services give us access to vast libraries of audio and video content that we consume with solitary enjoyment.

In my view, Shirky overestimated how much people, campaigning aside, really do want to come together. In the end, most technology, be it vacuum cleaners or voice-activated assistants, makes it easier for individuals to do what they want.

ABOUT THE AUTHOR

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

This article originally appeared in
Computing in Science & Engineering, *vol. 20, no. 2, 2018.*

Editor: **Erik Cambria,** Nanyang Technological University, Singapore, cambria@ntu.edu.sg

Aspect-Based Extraction and Analysis of Affective Knowledge from Social Media Streams

Albert Weichselbraun, *University of Applied Sciences Chur and webLyzard*
Stefan Gindl, *MODUL University Vienna*
Fabian Fischer and Svitlana Vakulenko, *Vienna University of Economics and Business*
Arno Scharl, *MODUL University Vienna and webLyzard*

This article introduces an approach to analyze emotional values associated with brands and companies. Online media coverage about products and services typically refers to a wide range of aspects to which such emotional values apply. These aspects can include product features (such as a digital camera's maximum resolution), common applications (such as a smartphone used as a car navigation system), or perceptions in conjunction with a specific event (for example, as part of a sponsorship agreement). Our approach integrates affective and factual knowledge extraction to capture opinions related to specific aspects along multiple emotional dimensions. We use the automotive industry as a sample domain to demonstrate the proposed approach, given the large number of aspects that characterize its complex technical products.

Affective knowledge includes sentiment and other emotions expressed in a document, which are captured and evaluated by opinion-mining algorithms. Typically, such algorithms are based on machine learning, lexical methods, or a combination of both.[1] To identify entities and aspects, the presented system also extracts *factual knowledge* using a knowledge base built on data from linked data sources such as DBpedia and ConceptNet. This knowledge base holds information about products, including not only product characteristics but also corporate decision makers such as Martin Winterkorn, the former CEO of Volkswagen AG (www.dbpedia.org/page/Martin_Winterkorn).

The real-time social media streams used for the analysis originate from the Media Watch on Climate Change (www.ecoresearch.net/climate), a continuously updated knowledge repository on climate change and related environmental issues.[2] The system is based on the webLyzard Web intelligence platform (www.weblyzard.com), which extracts and visualizes knowledge from digital content streams to measure the impact of events and communication campaigns, independent of a specific domain. Adapted to the specific requirements of the Media Watch on Climate Change, the system collects, filters, and annotates documents from news media, social networking platforms, and the websites of Fortune 1000 companies and environmental organizations.[3]

Figure 1 shows the results of a sample query for the term "Volkswagen" in English-language news media published between July and December 2016. The screenshot reflects the significant media impact of the "Dieselgate" scandal (that is, manipulations to cheat official pollution tests), with most of the articles about Volks-wagen still focusing on this story. The event's dominance highlights the importance of aspect-centered approaches to opinion mining. Although the overall sentiment is negative, specific features such as *seat quality* or the *gearbox* receive positive feedback. Only a granular analysis that considers all relevant aspects can reveal such hidden knowledge, which is highly relevant for planning and evaluating corporate communication campaigns.

We tackle this challenge using the four emotional categories of SenticNet[4] in addition to the standard sentiment polarity, which helps to distinguish different aspects of the target's emotional load, and computing per-aspect sentiment values that account for different properties relevant to users. The major challenge lies in identifying these relevant aspects. Most aspect-oriented sentiment analysis approaches

 Published by the IEEE Computer Society 2469-7087/18/$33.00

Figure 1. Screenshot of the Media Watch on Climate Change, a public Web intelligence application available at www.ecoresearch.net/climate (query: "Volkswagen"; English-language news media sites, 07–12/2016).

use frequency- and syntax-based methods without linking to background knowledge for further reasoning.

For evaluation purposes, we asked domain experts to assess the retrieved common and commonsense knowledge (for example, aspect is relevant, not relevant, or unsure whether it is relevant), and provided a radar chart visualization to demonstrate how the entities and the corresponding aspects perform according to the SenticNet emotional categories.

Methodology

Our approach pursues a flexible and automated strategy by linking input entities to DBpedia to obtain background knowledge on relevant properties of these entities (aspects). The dependency graph enrichment adds background information on emotional categories and trigger terms, as well as sentiment targets and aspects obtained from the knowledge acquisition components, yielding the opinion graph used in the knowledge extraction process. A text document is represented by several of these opinion graphs. Sentiment-target linking uses a machine learning classifier to connect sentiment targets and aspects to trigger terms. The component relies on sentence dependency graphs as input, which represent tokens as nodes and their dependencies as directed edges. The sentiment-parsing component finally extracts affective knowledge from the opinion graph. It refines this knowledge with factual knowledge on relevant sentiment targets and aspects obtained from graph mining, and stores it in the affective knowledge repository. Figure 2 summarizes the affective knowledge extraction process for identifying beliefs, opinions, and arguments in text documents.

Knowledge Acquisition

The knowledge acquisition component provides information for enriching dependency graphs, as outlined in the next section, with information on a term's polarity obtained from a polarity lexicon, its SenticNet emotional categories, common knowledge acquired from DBpedia, and commonsense knowledge from ConceptNet.

Since our work focuses on car brands and models, we only mine sentiment targets and sentiment aspects relevant to this domain. Algorithm 1 (Figure 3) captures information on companies, products, and aspects from DBpedia and ConceptNet. It obtains relations that lead from the entity (for example, "Volkswagen") to an associated aspect. For instance, the relation "manufacturer" yields "Lupo" or "Golf" from DBpedia. This association reveals that Volkswagen manufactured the car models

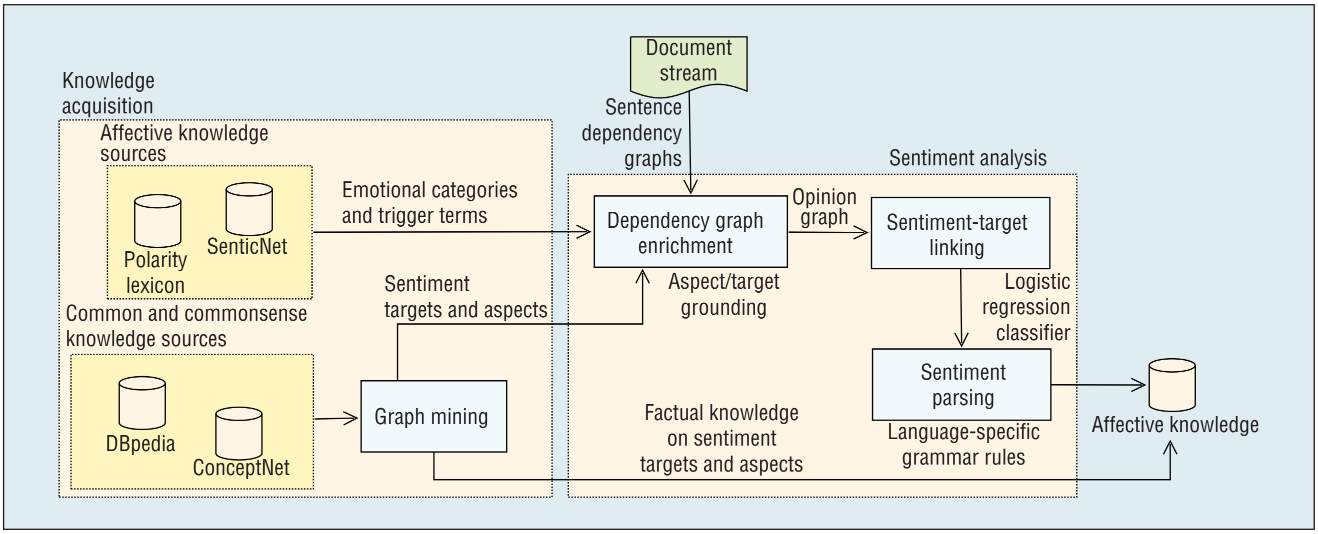


Figure 2. Main components of the affective knowledge extraction process. Preprocessing transforms documents into dependency graphs that are then enriched with external knowledge obtained from the knowledge acquisition component to create opinion graphs. Sentiment analysis extracts affective knowledge from these graphs that is then combined and extended with common and commonsense knowledge.

```
Require: sets of target_industries, target_predicates and product_predicates
// Lists for storing the results of the graph mining process
companies ←{}, products ←{}, entity_graph ←{},
// graph mining
for all triple from query (?s <rdf:type> <dbo:Company>) do
 if (?s <dbp:industry> ?o) and ?o in target_industries then
  companies.add(triple.s)
  entity_graph.add_triple(triple)
 end if
end for
for all triple from (query (?s ?p ?o ∈companies)
         ∪query(?s ∈companies ?p ?o)
         ∪query(?s ?p ∈target_predicates ?o))
do
 if triple.p ∈product_predicates then
  products.add(triple.o)
 end if
 entity_graph.add_triple(triple)
end for
for all triple from query (?s <dbp:aka> ?o) do
 entity_graph.add_triple(triple)
end for
return companies, products, entity_graph
```

Figure 3. Algorithm 1 extracts sentiment targets and aspects as well as the corresponding context information from DBpedia. At first, the algorithm mines companies and products relevant to the target industry and then obtains subgraphs with context information on these two entity types.

Lupo and Golf. The DBpedia relation "keyPerson" yields "Martin Winterkorn" and "Ferdinand Piëch," both former chairs of Volkswagen, as important persons related to the company.

A set of predefined relations helps restrict the aspects to those most relevant for the investigation:

- DBpedia: dbo:manufacturer, dbo:keyPerson, dbo:product, dbp:team
- ConceptNet: PartOf, HasA, UsedFor, MadeOf

The algorithm obtains not only the label of the DBpedia resources but also all linked aliases. Additionally, it automatically creates aliases by removing tokens that are shared between the manufacturer and the product. This means it can automatically create the alias "Golf" from the car entity "Volks-wagen Golf" and the company entity "Volkswagen," thereby increasing the achievable recall.

To increase coverage, the graph-mining component queries ConceptNet for automobile properties and adds commonsense knowledge, such as that a car is a means of transport and has a steering wheel and a trunk (aspects). This component later uses the obtained relations (such as products produced by a company or key people working for that company) in conjunction with commonsense knowledge (such as major parts of such an entity or its typical applications) to enrich the dependency graph.

This knowledge-rich approach has two advantages over frequency-based methods that rely on syntactic features. First, the created affective knowledge base captures not only related entities but also the corresponding relation types. Second, grounding targets to DBpedia helps to obtain additional information such as abstracts, further relations, and car type.

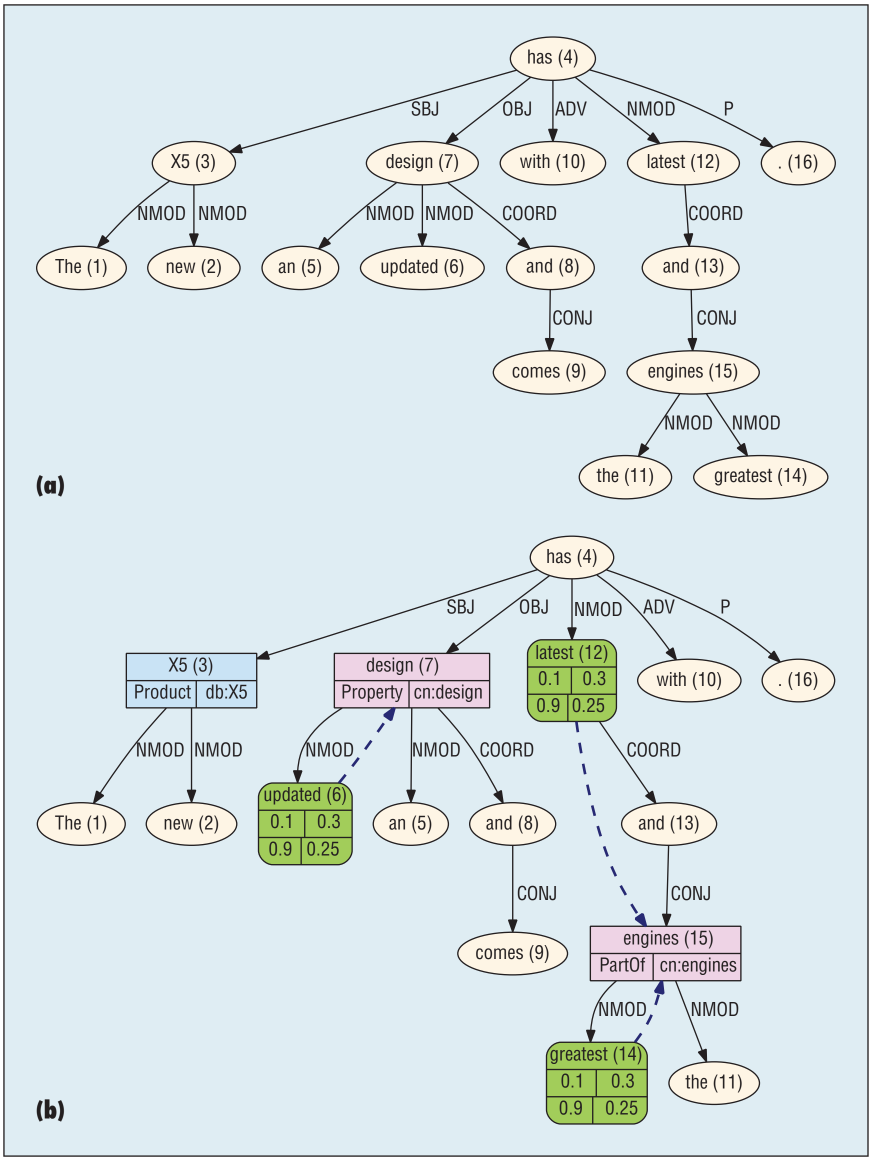


Figure 4. Dependency graph enrichment: (a) dependency tree of the sentence, "The new X5 has an updated design and comes with the latest and greatest engines"; and (b) enriched with opinions (blue: targets with type and DBpedia concept; violet: aspects with type and ConceptNet grounding; green: positive sentiment terms and sentic values; and dashed lines connect sentiment terms with their targets).

Sentiment Analysis

Using the affective and factual resources provided by the knowledge acquisition component, sentiment analysis follows a three-step process: dependency graph enrichment, sentiment-target linking, and sentiment parsing.

Dependency graph enrichment. Enriching the sentence dependency graphs with emotional categories, trigger terms that indicate negations or modify sentiment values, sentiment targets, and sentiment aspects obtained from the knowledge acquisition component yields the *opinion graph*, which we use in the subsequent sentiment-target linking and sentiment parsing steps.

After creating the dependency parse tree (see Figure 4a), the system draws upon the knowledge acquisition component to ground target concepts

Related Work in Sentiment Analysis

A better understanding of sentiment is crucial for building next-generation artificial intelligence systems and increasing the value of business intelligence applications.[1] This requires the integration of multiple approaches into a unified system, including the three research areas outlined in the following.

Emotion Analysis

Emotion analysis draws upon psychology research. For instance, SenticNet[2] is based on Plutchik's wheel of emotions.[3] It contains 50,000 concepts and maps them to the four dimensions proposed in the Hourglass of Emotions[4]: "aptitude" (confident in interaction benefits), "attention" (interested in interaction contents), "pleasantness" (amused by interaction modalities), and "sensitivity" (comfortable with interaction dynamics). WordNet-Affect[5] has affective labels such as "emotion," "mood," and "cognitive state" to approximately 2,800 WordNet synsets. The General Inquirer provides emotional categories such as "virtual," "pleasure," and "pain."[6] EmoLex contains approximately 10,000 terms,[7] and Affective Norms for English Words knows the three categories "valence" (from unpleasant to pleasant), "arousal" (from calm to excited), and "dominance."[8]

Sentiment-Target Linking

This research field identifies the target of an opinionated statement. For instance, "VW Golf" is the target of "reliable" in the statement, "The VW Golf is reliable." Rule-based approaches to sentiment-target linking use manually designed heuristics to find valid sentiment-target pairs—for example, sentiment-target proximity (distance-based approaches),[9] semantic frames,[10] or syntax-based approaches relying on a handful of patterns.[11,12] Supervised machine learning methods collect patterns from annotated corpora automatically. For example, Lei Zhuang and his colleagues[13] and Liheng Xu[14] automatically extract dependency patterns between sentiments and their targets.

Corpora such as J.D. Power and Associates (JDPA) support the evaluation of such tools.[15] We used a similar approach to build our classifier and further optimized its performance by evaluating and selecting features and including additional patterns learned from the multiperspective question answering (MPQA) corpus.[16]

Aspect-Based Sentiment Analysis

Aspect-based sentiment analysis extends target-dependent sentiment analysis and identifies opinions on aspects of that entity. For example, given an entity "car," its design and engine characteristics are different aspects of the same entity. Most research focuses on product reviews and links mentioned aspects to opinions.[17] State-of-the-art approaches use term or *n*-gram frequencies[18,19] and frequently employ machine learning—for example, conditional random fields (CRF),[20] deep learning,[21] and latent Dirichlet allocation (LDA).[22] Other approaches combine syntactic rules and lexical resources.[23,24]

Our approach uses a knowledge base to identify aspects. This approach is similar to work by Caroline Brun and her colleagues, who bootstrap an aspect lexicon using a training corpus by combining WordNet and Wikipedia,[25] or Basant Agarwal and his colleagues, who access ConceptNet and WordNet to create a product-review-specific ontology.[26]

Proper opinion analysis is a combination of all these methods. After identifying an emotion, it is necessary to connect it to its target to allow reasoning such as, "who thinks what about whom?" Finally, identifying additional aspects related to the target gives higher granularity and further insight into the true meaning of the expressed opinion.

(that is, cars) to DBpedia. Afterward, it uses this information together with the context retrieved from DBpedia to query the knowledge acquisition component for aspects relevant to the targets from ConceptNet and to link these aspects to the corresponding ConceptNet nodes (see Figure 4b).

The affective knowledge extraction uses lexical lookups to identify tokens carrying affective knowledge and assigns them a value in the range [–1, 1]. The component supports multiple emotional categories. Grounding emotion triggers is not limited to string matching; rather, it is also aware of parts-of-speech (POS) tags. In the case of "like," for example, it differentiates between the use as a positive verb and as a neutral comparison term.

The system ignores product aliases unless the entity (obtained from DBpedia), its manufacturer, or the company's aliases occur in the text. This avoids problems with generic names (such as numbers, frequent domain-agnostic terms, or short character sequences) and allows it to correctly identify "BMW" and "X5" (for example, in "Yesterday BMW showed its newest SUV for the first time. The new X5 has an updated design and comes with the latest and greatest engines") without creating links if "BMW" is not mentioned.

The discovery of an aspect requires its subsequent linking to an entity (for example, "steering wheel" and "car"). A colocation heuristic helps find the closest candidate by scanning the current sentence first and, if unsuccessful, the entire document. The sentiment-target linking classifier then links the common and commonsense knowledge to the affective knowledge targeted at it.

Sentiment-target linking. Sentiment-target linking uses a set of sentiment terms (that is, terms indicating a certain emotion or sentiment) $S_m = \{t_{s_i}\}$ and target terms (that is, sentiment targets or aspects) $T_m = \{t_{t_j}\}$ extracted from sentence m, and returns a set of valid sentiment-target pairs: $\{(t_{s_i}, t_{t_j})\}$,

References

1. E. Cambria, "Affective Computing and Sentiment Analysis," *IEEE Intelligent Systems*, vol. 31, no. 2, 2016, pp. 102–107.
2. E. Cambria and A. Hussain, *Sentic Computing: A Common-Sense-Based Framework for Concept-Level Sentiment Analysis*, Springer, 2015.
3. R. Plutchik, "The Nature of Emotions," *Am. Scientist*, vol. 89, no. 4, 2001, pp. 344–350.
4. E. Cambria, A. Livingstone, and A. Hussain, "The Hourglass of Emotions," *Cognitive Behavioural Systems*, LNCS 7403, Springer, 2012, pp. 144–157.
5. C. Strapparava and A. Valitutti, "WordNet-Affect: An Affective Extension of WordNet," *Proc. 4th Int'l Conf. Language Resources and Evaluation* (LREC 04), 2004, pp. 1083–1086.
6. P.J. Stone, *The General Inquirer: A Computer Approach to Content Analysis*, MIT Press, 1966.
7. S. Mohammad and P.D. Turney, "Crowdsourcing a Word-Emotion Association Lexicon," *Computational Intelligence*, vol. 29, no. 3, 2013, pp. 436–465.
8. M.M. Bradley and P.J. Lang, *Affective Norms for English Words (ANEW): Stimuli, Instruction Manual, and Affective Ratings*, tech. report, Center for Research in Psychophysiology, Univ. of Florida, Gainesville, 1999.
9. M. Hu and B. Liu, "Mining and Summarizing Customer Reviews," *Proc. 10th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, 2004, pp. 168–177.
10. S.-M. Kim and E. Hovy, "Extracting Opinions, Opinion Holders, and Topics Expressed in Online News Media Text," *Proc. Workshop Sentiment and Subjectivity in Text*, 2006, pp. 1–8.
11. S. Gindl, A. Weichselbraun, and A. Scharl, "Rule-Based Opinion Target and Aspect Extraction to Acquire Affective Knowledge," *Proc. WWW Workshop Multidisciplinary Approaches to Big Social Data Analysis* (MABSDA 13), 2013, pp. 557–564.
12. S. Poria et al., "Sentiment Data Flow Analysis by Means of Dynamic Linguistic Patterns," *IEEE Computational Intelligence*, vol. 10, no. 4, 2015, pp. 26–36.
13. L. Zhuang, F. Jing, and X.-Y. Zhu, "Movie Review Mining and Summarization," *Proc. 15th ACM Int'l Conf. Information and Knowledge Management*, 2006, pp. 43–50.
14. L. Xu et al., "Mining Opinion Words and Opinion Targets in a Two-Stage Framework," *Proc. 51st Ann. Meeting Assoc. for Computational Linguistics* (ACL 13), vol. 1, 2013, pp. 1764–1773.
15. J.S. Kessler and N. Nicolov, "Targeting Sentiment Expressions through Supervised Ranking of Linguistic Configurations," *Proc. 3rd Int'l AAAI Conf. Weblogs and Social Media* (ICWSM 09), 2009, pp. 90-97.
16. T. Wilson, "Fine-Grained Subjectivity and Sentiment Analysis: Recognizing the Intensity, Polarity, and Attitudes of Private States," PhD diss., Univ. of Pittsburgh, 2008.
17. M. Pontiki et al., "Semeval-2014 Task 4: Aspect-Based Sentiment Analysis," *Proc. 8th Int'l Workshop Semantic Evaluation* (SemEval 14), 2014, pp. 27–35.
18. E. Marrese-Taylor, J.D. Velásquez, and F. Bravo-Marquez, "A Novel Deterministic Approach for Aspect-Based Opinion Mining in Tourism Products Reviews," *Expert Systems with Applications: An Int'l J.*, vol. 41, no. 17, 2014, pp. 7764–7775.
19. Y. Wu and M. Ester, "FLAME: A Probabilistic Model Combining Aspect Based Opinion Mining and Collaborative Filtering," *Proc. 8th ACM Int'l Conf. Web Search and Data Mining* (WSDM 15), 2015, pp. 199–208.
20. M. Chernyshevich, "Cross-Domain Extraction of Product Features Using Conditional Random Fields," *Proc. 8th Int'l Workshop on Semantic Evaluation* (SemEval 14), 2014, p. 309–313.
21. S. Poria, E. Cambria, and A. Gelbukh, "Aspect Extraction for Opinion Mining with a Deep Convolutional Neural Network," *Knowledge-Based Systems*, vol. 108, Special Issue on New Avenues in Knowledge Bases for Natural Language Processing, 2016, pp. 42–49.
22. S. Poria et al., "Sentic LDA: Improving on LDA with Semantic Similarity for Aspect-Based Sentiment Analysis," *Proc. Int'l Joint Conf. Neural Networks* (IJCNN 16), 2016, pp. 4465–4473.
23. K. Veselovská and A. Tamchyna, "UFAL: Using Hand-crafted Rules in Aspect Based Sentiment Analysis on Parsed Data," *Proc. 8th Int'l Workshop Semantic Evaluation* (SemEval 14), 2014, pp. 694–698.
24. A.G. Pablos et al., "Unsupervised Acquisition of Domain Aspect Terms for Aspect Based Opinion Mining," *Procesamiento del Lenguaje Natural*, vol. 53, Sept. 2014, pp. 121–128.
25. C. Brun, D.N. Popa, and C. Roux, "XRCE: Hybrid Classification for Aspect-Based Sentiment Analysis," *Proc. 8th Int'l Workshop Semantic Evaluation* (SemEval 14), 2014, pp. 838–842.
26. B. Agarwal et al., "Sentiment Analysis Using Common-Sense and Context Information," *J. Computational Intelligence and Neuroscience*, vol. 2015, Jan. 2015, article 30.

where $y\left(t_{s_i}, t_{t_j}\right) = True$. Hereby, we formulate the sentiment-target linking task as a binary classification problem. The classification function y reflects whether sentiment t_{s_i} and target tokens t_{t_j} constitute a valid sentiment-target pair.

The component starts by generating all possible edges between the set of targets and the set of sentiments as candidates for valid sentiment-target pairs and further evaluates each of them independently.

The component extracts features for every observation of a sentiment-target pair and uses them as input for the classification model previously trained on a corpus annotated with correct sentiment-target pairs. To train the classifier, it uses observations from a corpus annotated with words and phrases expressing sentiments $\left\{t_{s_i}\right\}$, targets $\left\{t_{t_j}\right\}$, and relations between them $\left\{\left(t_{s_k}, t_{t_l}\right)\right\}$. An observation $x\left(t_{s_i}, t_{t_j}\right)$ is a set of features that captures syntactic relations between the sentiment token t_{s_i} and the target token t_{t_j}. A recursive feature elimination (RFE) procedure yields an optimal feature set to be extracted from the opinion graph for each observation of a sentiment-target pair $x\left(t_{s_i}, t_{t_j}\right)$, which comprises features such as POS tags and dependencies between the sentiments and target nodes in the graph.

The sentiment-target linking uses a logistic regression classifier trained on the J.D. Power and Associates (JDPA, http://verbs.colorado.edu/jdpacorpus) sentiment corpus and the Multiperspective Question Answering (MPQA, http://mpqa.cs.pitt.edu/corpora/mpqa_corpus) opinion corpus, version 2.0 (also see the sidebar). An evaluation of the sentiment-target linking performance achieved an F-measure of 0.90 when evaluated on the gold-standard annotations for about 12,000 sentiment-target pairs with stratified tenfold cross validation.

Sentiment parsing. Grammar rules and heuristics help identify and extract

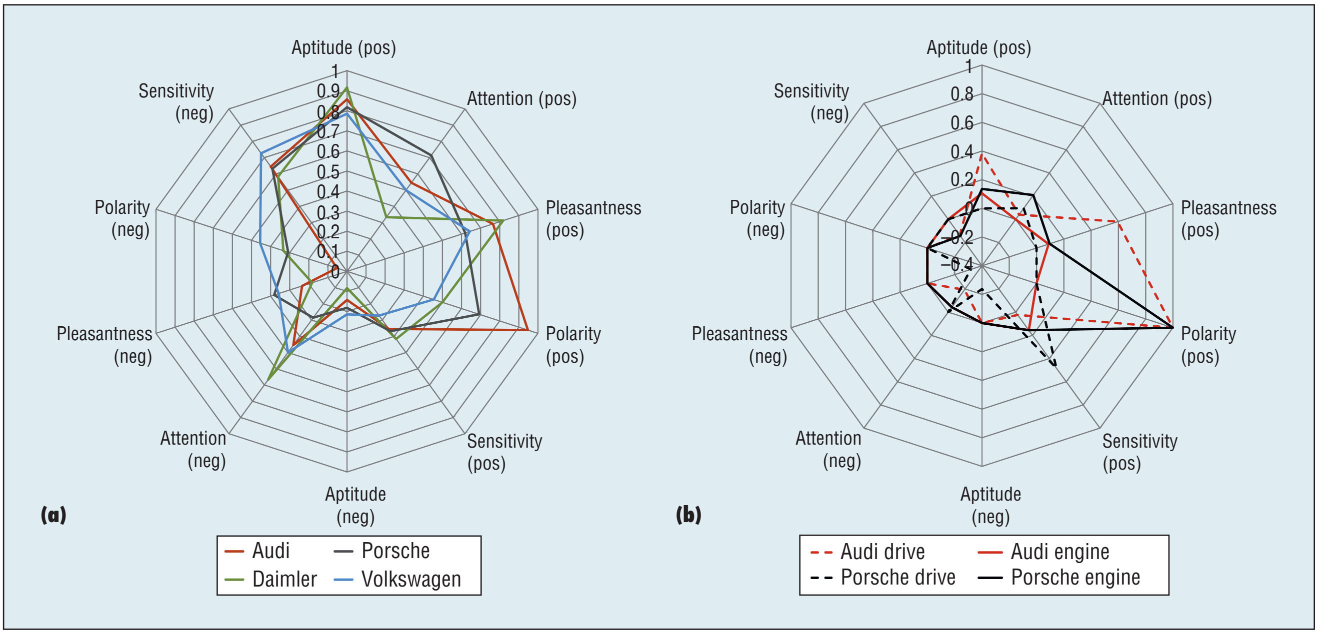


Figure 5. SenticNet emotional categories and polarities for (a) selected car brands, and (b) the aspects (in this case, product features) "drive" and "engine."

Table 1. Statistics of the acquired background knowledge.

Description	No.
Companies active in the automotive industries	881
Key people in these companies	349
Car entities	4,898
Car aliases	7,111
Car aspects	30

affective knowledge—for example, negation detection to invert the polarity of a negated term. It uses nodes marked as triggers and stoppers to determine the start and end of the negation scope within the opinion graph, and supports multiple negation.

Aggregating the opinion triggers that have been linked to a particular sentiment target yields the target's value for the corresponding emotional category. By considering different sentiment aspects in this aggregation process, the system can analyze the emotions contributed by each aspect, yielding visualizations such as the one presented in Figure 5.

Data Analytics

An RDF triple store serves to store affective and factual knowledge. A proof-of-concept data analytics application queries the affective knowledge base to compare the emotions associated with four automobile brands having high media coverage (Audi, Daimler, Porsche, and Volkswagen). It contrasts this analysis with an evaluation of two different aspects (drive and engine) relevant to products of two of these brands (Audi and Porsche).

The affective knowledge repository facilitates polarity classification and emotional analysis aligned with the "Hourglass of Emotions" (see the sidebar). For instance, the "engine" of "VW" receives a sensitivity of −0.07, whereas "Golf" has a sensitivity of 0.014. After determining the emotional strength associated with each company and aspect, we aggregate over all aspects and calculate a total value using the following formula:

$$strength_{emotion} = \frac{k}{n}, \qquad (1)$$

where k is the number of positive occurrences (negative occurrences for negative strength) of the emotional dimension, while n is the total number of occurrences of this emotion. A summary of the obtained results is presented later.

Experiments

Using a subset of the archive of the Media Watch on Climate Change (social media messages published between 28 September and 28 November 2015), the evaluation corpus consists of 1,000 Twitter and Google+ postings containing the word "car," and 4,000 referring to one of the car brands Audi, Daimler, Porsche, and Volkswagen. The former helped extract sentiment aspects and targets contained in the knowledge base, the latter supported the evaluation of aspect-based emotion analysis.

Graph Mining Results

The approach introduced earlier yields a considerable amount of background knowledge from DBpedia and ConceptNet that has been used for the sentiment analysis. Table 1 lists the number

Table 2. Extracted entities and aspects connected to the car company Tesla Motors and the car Tesla Model S.

Entity	Relation	Aspect
Tesla Motors	Type	Company
	Industry	Car
	Manufacturer	Tesla Model S, Tesla Roadster
	Product	Luxury vehicle
	Key person	JB Straubel, Elon Musk, chief executive officer, chief technology officer, chair
Tesla Model S	Aka	WhiteStar, Model S
	HasA	trunk, radio, headlight, four wheel, seat, wheel, engine, window, four tires
	MadeOf	steel, metal
	PartOf	trunk, engine, transmission, radiator, body, hood, tire, fender, door, tire, engine, steer wheel, drive train, wheel
	UsedFor	drive, transportation, travel

of entities, aliases, and aspects acquired from the common and commonsense knowledge sources.

Table 2 shows the obtained entities and properties for the company Tesla Motors and the car Tesla Model S, demonstrating the level of detail achieved with the presented approach.

Evaluation of the Extracted Knowledge

The following evaluation draws upon the 50 most frequently occurring sentiment aspects and targets in the evaluation corpus to assess the usefulness and impact of the knowledge extracted by the graph mining. Five independent domain experts classified the usefulness of each extracted concept for describing aspects relevant to the perception (polarity, emotions) of car companies, brands, and products in one of three categories: *useful* (the aspect is related to the domain), *not useful* (the aspect has no connection to the domain), and *neutral* (the term is too generic to be clearly associated with the domain). On average, 81.2 percent of the extracted concepts have been considered useful. The Krippendorff alpha for inter-rater agreement between experts is 0.504, reflecting only a moderate agreement among domain experts.

The evaluation illustrates two shortcomings of the current approach. First, the assumption that automotive companies only manufacture cars does not hold true. Among the 50 most frequent entities/aspects in the "car" corpus, the system identified "knife" because the company American Expedition Vehicles also produces knives. We investigated narrowing the products based on their `rdfs:class` property but encountered a diverse set of assigned classes that have no single common superclass or shared property.

Second, two ambiguous car brands showed up in the evaluation: the short-lived WiLL and SEAT, which was often confused with car seat. WiLL could be tackled by allowing only certain aspects to be matched with verbs (for example, aspects connected with the "UsedFor" predicate to a car). SEAT, however, is difficult to ground: in social media, capitalization cannot reliably be used for disambiguation (since often the text is all lowercase), and the domain car fits both car seat and the car brand SEAT.

Aspect-Based Analysis of Brand Perceptions

Using the data analytics approach presented earlier and building on previous work to visualize emotions along multiple dimensions,[5] we show how the affective knowledge extracted from social media messages can be associated with the investigated car brands (Figure 5a). Applying the Hourglass of Emotions to the emotional dimensions "aptitude," "attention," "pleasantness," and "sensitivity" lets us map numerical chart values to their emotional equivalents. The car brand Audi, for example, shows a strong association with positive aptitude (0.86), which maps to the emotion "admiration" on the Hourglass of Emotions. The brand is also associated with a moderate negative sensitivity (0.65), which is equivalent to "fear." Negative attention (0.45) reveals that "surprise" is also associated with "Audi."

"Volkswagen" has the most significant peaks in the negative direction—for example, a negative attention of 0.5, a negative pleasantness of 0.36, and a negative sensitivity of 0.72. These values map to "surprise," "sadness," and "terror" on the Hourglass of Emotions. This result is in line with the negative media coverage about the exhaust scandal.

Emotion analysis provides detailed feedback on the public perception of a company. A brand might outperform another in one aspect, such as product quality, but might have to catch up on another aspect, such as service. The radar chart in Figure 5b, for example, shows that the Porsche engine has a considerably higher attention and sentiment than its competitor, but Audi excels in pleasantness and sentiment when focusing on actually driving the car.

Among the main challenges of deploying aspect-based opinion mining algorithms for Web intelligence applications are the required scalability of the computational methods, and appropriate visual representations that convey the aspect structure and associated emotions in an intuitive manner. The European research project Adaptive Scalable Analytics Platform (ASAP, www.asap-fp7.eu) is currently tackling both challenges. ASAP will enable us to perform the required complex computations on high-volume content streams from social networking platforms, and to provide real-time visualizations of the evolving aspect structure as part of an interactive dashboard—going beyond standard representations such as trend lines and radar charts.[6]

Future research will also apply the presented methods in different domains and demonstrate their applicability beyond specific products and services. Measuring the impact of international marketing and public outreach campaigns, for example, would significantly benefit from an aspect-oriented approach. Simple bipolar metrics such as sentiment cannot adequately reflect the underlying complexities when millions of stakeholders use digital channels to participate in public debates about complex, multi-faceted topics. ■

Acknowledgments

The work presented here was conducted as part of the IMAGINE project, funded by the Swiss Commission for Technology and Innovation (CTI); the uComp project, which receives funding from EPSRC EP/K017896/1, FWF 1097-N23, and ANR-12-CHRI-0003-03, in the framework of CHIST-ERA ERA-NET; and the DecarboNet and ASAP projects, which receive funding from the EU 7th Framework Program for Research, Technology Development and Demonstration under grant agreements 610829 and 619706, respectively.

References

1. A. Weichselbraun, S. Gindl, and A. Scharl, "Extracting and Grounding Contextualized Sentiment Lexicons," *IEEE Intelligent Systems*, vol. 28, no. 2, 2013, pp. 39–46.
2. A. Scharl et al., "Semantic Systems and Visual Tools to Support Environmental Communication," *IEEE Systems J.*, preprint, 8 Sept. 2015; doi:10.1109/JSYST.2015.2466439.
3. A. Scharl et al., "Scalable Knowledge Extraction and Visualization for Web Intelligence," *Proc. 49th Hawaii Int'l Conf. System Sciences* (HICSS 16), 2016, pp. 3749–3757.
4. E. Cambria et al., "SenticNet 4: A Semantic Resource for Sentiment Analysis Based on Conceptual Primitives," *Proc. 26th Int'l Conf. Computational Linguistics: Technical Papers* (COLING 16), 2016, pp. 2666–2677.
5. A. Scharl et al., "Analyzing the Public Discourse on Works of Fiction—Automatic Emotion Detection in Online Media Coverage about HBO's Game of Thrones," *Information Processing and Management*, vol. 52, no. 1, 2016, pp. 129–138.
6. A.M. Brasoveanu et al., "Visualizing Statistical Linked Knowledge for Decision Support," *Semantic Web J.*, vol. 8, no. 1, 2017, pp. 113–137.

Albert Weichselbraun is a professor of information science in the Swiss Institute for Information Research at the University of Applied Sciences Chur, and technical director of webLyzard technology. His research interests include big data, natural language processing, and text mining. Weichselbraun has a habilitation and doctorate in information systems from the Vienna University of Economics and Business. Contact him at albert.weichselbraun@htwchur.ch.

Stefan Gindl is an assistant professor in the Department of New Media Technology at MODUL University Vienna. His research interests include text mining, natural language processing, machine learning, and artificial intelligence. Gindl has a PhD in computer sciences from the Vienna University of Technology. Contact him at stefan.gindl@modul.ac.at.

Fabian Fischer is a researcher in the Institute for Information Business at the Vienna University of Economics and Business. His research interests include natural language processing, algorithmics, big data, and privacy. Fischer has an MS in computational intelligence from the Vienna University of Technology. Contact him at fabian.fischer@wu.ac.at.

Svitlana Vakulenko is a researcher in the Institute for Information Business at the Vienna University of Economics and Business. Her research interests are natural language processing, including information extraction, machine learning, and social media. Vakulenko has an MSc in software engineering from the University of Tartu, Estonia. Contact her at svitlana.vakulenko@wu.ac.at.

Arno Scharl is a professor of information systems in the Department of New Media Technology at MODUL University Vienna, and the managing director of webLyzard technology. His research interests include media intelligence, big data analytics, information visualization, and the integration of semantic and geospatial Web technology. Scharl has a habilitation and doctorate in management information systems from the Vienna University of Economics and Business. Contact him at scharl@weblyzard.com.

This article originally appeared in IEEE Intelligent Systems, *vol. 32, no. 3, 2017.*

The Rise of Intelligent Cyber-Physical Systems

Hausi A. Müller, University of Victoria

It's expected that the cyber-physical systems revolution will be more transformative than the IT revolution of the past four decades.

Cyber-physical systems (CPS) are orchestrations of computers, machines, and people working together to achieve goals using computation, communications, and control (CCC) technologies. Although the term CPS was coined only in 2006 by Helen Gill of the National Science Foundation (NSF), the CCC core technologies of CPS have had a rich and long history. Major milestones for CPS include control theory in 1868, wireless telegraphy in 1903, cybernetics feedback in 1948, embedded systems in 1961, software engineering in 1968, and ubiquitous computing in 1988. CPSs have risen from the field of embedded systems to the realm of digital ecosystems and are becoming increasingly intelligent as a result of analytics and machine-learning capabilities being readily available in the cloud and accessible over networks. The advances in the interconnected capabilities of CPSs affect virtually every engineered system and will enable adaptability, scalability, resiliency, safety, security, and usability in future CPSs that will far exceed the systems of today.

Over the past two decades, the number of cyber components has grown gradually to the point where CPSs are now software-intensive systems with more and more integrated computing hardware and computational algorithms. In today's CPS, software dominates all aspects of connecting the physical and cyber worlds by orchestrating the CCC technologies in CPS applications. Consequently, the engineering of high-confidence CPSs has also evolved. The resulting process is neither an extension of traditional engineering nor a straightforward application of software engineering,[1] but rather a new systems engineering science. Granting agencies around the world have recognized this problem and initiated large research programs to investigate CPS foundations. A key goal of the NSF CPS research program is to develop the core systems science needed to engineer complex CPSs. The idea is to abstract from specific systems and application domains to reveal fundamental CPS engineering principles.

Over the years, engineers have been highly successful in developing models for specific control system applications. Integrating discrete, continuous, and adaptive control as well as deterministic and nondeterministic models are fundamental challenges in dealing with uncertainty in modern CPSs. Developing models and modeling frameworks for CPS has become a mature research field.[2–4] The software engineering community has made tremendous strides in designing and operating highly dynamical software systems by developing methods and techniques to standardize and distribute CPS components and services effectively through autonomic computing[5] (for example, the Monitor-Analyze-Plan-Execute loop operating on a shared Knowledge [MAPE-K] base), to control feedback in computing systems,[6] to deal with inherent uncertainty in CPS through models at runtime, and to adapt and then validate CPS at runtime. Several research communities have emerged to deal with software engineering aspects of CPS, including CPS conferences and workshops (such as CPS Week), software engineering for adaptive and self-managing systems (SEAMS),[7] Models@run.time,[8,9] as well as runtime validation, verification, and certification techniques.[10]

For the past decade, think tanks and granting agencies (such as NSF, NIST, the National Institutes of Health [NIH], EU Horizon 2020, and Europe 2020) have articulated their vision on the future of CPS applications. Their tenor is similar: the expectation is that the CPS revolution will be more transformative than the IT revolution of the past four decades.[11,12]

Why is this CPS revolution happening now? The primary reason is the recent confluence of technologies, including adaptive systems and runtime models, an increasingly instrumented world due to pervasive sensing and actuating capabilities, advanced real-time and networked control, analytical and cognitive capabilities, and compute and storage clouds. With the advent of cognitive intelligent assistants readily available on personal devices, human-in-the-loop CPSs are proliferating in our lives.[13,14] In other words, CPS is at the center of a perfect technology storm. Countries around the world are investing heavily in CPS research programs, seeking a technological and economic edge.[1]

There are several terms and fields closely related and competing with the notion of CPS, including embedded systems, the Internet of Things (IoT), the Industrial Internet (II), the Internet of Everything (IoE), machine-to-machine (M2M), Industry 4.0, Smarter Planet, cyber-physical-human systems (CPHS), smart and intelligent systems, and adaptive systems. While all these fields have their own publications and communities, UC Berkeley professor Edward A. Lee argues convincingly that the CPS term is more foundational and encompassing than these related terms, because the term embodies the fundamental engineering problem of integrating the cyber and physical worlds.[2]

CPS is at the center of a perfect technology storm.

There are many challenges that must be addressed to be able to harvest CPS's rich economic opportunities. As Sir Francis Bacon said, "If we are to achieve results never before accomplished, we must expect to employ methods never before attempted."

First and foremost, creating and maintaining a skilled workforce to support the design, engineering, deployment, and operation of future CPS is a significant challenge for industry, academia, and governments. CPS engineers, scientists, and developers need not only strong backgrounds in CCC, but also significant knowledge in relevant application domains. Existing engineering and computer science programs are challenged in teaching the comprehensive skills required for a successful career in the CPS realm. Urgently, computer science and software engineering programs need to require control engineering courses, and traditional engineering programs need to include advanced software engineering courses.

CPS technologies are becoming the key enablers for building smarter infrastructures for industrial applications. Growing human populations consume enormous natural resources and require increasingly instrumented and optimized food supply chains. Flourishing cities require renewable energy systems and instrumented transportation infrastructure. Connected and autonomous vehicles combine situational awareness in vehicles with the networked infrastructure of the modern city. Rising costs put pressure on healthcare and elder care, requiring outcome prediction based on improved diagnostics using smart medical devices. Assistive healthcare systems—including wearable sensors, implantable devices, and home monitoring systems—are being developed to improve outcomes and quality of life. Thus, the technologies and applications emerging from combining the cyber and physical worlds will provide an innovation and incubation engine for a broad range of industries—creating entirely new markets and platforms for years to come. Our modern societies and economies increasingly depend on integrated, software-intensive CPS.

REFERENCES

1. M. Broy and A. Schmidt, "Challenges in Engineering Cyber-Physical Systems," *Computer*, vol. 47, no. 2, 2014, pp. 70–72.
2. E.A. Lee, "The Past, Present and Future of Cyber-Physical Systems: A Focus on Models," *Sensors*, vol. 15, no. 3, 2015, pp. 4837–4869.

3. E.A. Lee and S.A. Seshia, *Introduction to Embedded Systems: A Cyber-Physical Systems Approach*, 2nd ed., lulu.com, 2015.
4. E.A. Lee, "Fundamental Limits of Cyber-Physical Systems Modeling," *ACM Trans. Cyber-Physical Systems*, vol. 1, no. 1, 2017, article no. 3.
5. J.O. Kephart and D.M. Chess, "The Vision of Autonomic Computing," *Computer*, vol. 36, no. 1, 2003, pp. 41–50.
6. J.L. Hellerstein et al., *Feedback Control of Computing Systems*, Wiley-IEEE Press, 2004.
7. R. de Lemos et al., eds., *Software Engineering for Self-Adaptive Systems II*, LNCS 7475, Springer, 2013.
8. N. Bencomo et al., eds., *Models@run.time: Foundations, Applications, and Roadmaps*, Springer, 2014.
9. M. Szvetits and U. Zdun, "Systematic Literature Review of the Objectives, Techniques, Kinds, and Architectures of Models at Runtime," *Software & Systems Modeling*, vol. 15, no. 1, 2016, pp. 31–69.
10. S. Bhattacharyya et al., *Certification Considerations for Adaptive Systems*, tech. report NASA/CR–2015-218702, NASA, 2015; ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20150005863.pdf.
11. *Foundations for Innovation in Cyber-Physical Systems: Workshop Report*, tech. report, NIST, 2013; www.nist.gov/sites/default/files/documents/el/CPS-WorkshopReport-1-30-13-Final.pdf.
12. O. Vermesan and P. Friess, eds., *Internet of Things—Converging Technologies for Smart Environments and Integrated Ecosystems*, River Publishers, 2013.
13. G. Schirner et al., "The Future of Human-in-the-Loop Cyber-Physical Systems," *Computer*, vol. 46, no. 1, 2013, pp. 36–45.
14. S.K. Sowe et al., "Cyber-Physical-Human Systems: Putting People in the Loop," *IT Professional*, vol. 18, no. 1, 2016, pp. 10–13.

This article originally appeared in Computer, *vol. 50, no. 12, 2017.*

HAUSI A. MÜLLER is a professor of computer science and the Associate Dean of Research of the Faculty of Engineering at the University of Victoria. He is also the 2016–2018 vice president of Technical and Conferences Activities for the IEEE Computer Society. Contact him at hausimuller@gmail.com.

How Do You Create an Internet of Things Workforce?

Joanna F. DeFranco
Penn State University

Mohamad Kassab
Penn State University

Jeffrey Voas
IEEE Fellow

Editor:
Jeffrey Voas, NIST;
j.voas@ieee.org

Internet of Things (IoT) products and cyber-physical systems (CPS) are being utilized in almost every discipline. According to *Forbes*, there will be a significant increase in spending on the design and development of IoT applications and analytics. Furthermore, the most significant increase in spending will be in the business-to-business (B2B) IoT systems (such as manufacturing, transportation, and utilities), which is projected to reach $267 billion by 2020.[1] In addition to B2B, smart products are becoming more prevalent, such as thermostats, energy monitors, and light bulbs. Products that sense, learn, and react to user preferences are gaining popularity.

There are also CPS/IoT applications for healthcare with the goal of improving a patient's treatment regime. For example, the closed-loop insulin delivery system connecting a glucose monitor to an insulin pump can continuously alter the amount of insulin dosed to a patient to assist in managing the patient's blood sugar. In fact, any product that continuously monitors patient activity to improve treatment would be an effective IoT application. Imagine how much more effective treatment could be for a Parkinson's patient when a physician has more than a static snapshot from an office visit exam. With months of data and information, the physician could determine a more effective treatment plan. Accordingly, engineers and computer scientists also need the appropriate training to build safe and effective systems, whether part of the IoT or not. However, it is not sufficient to simply add one or two IoT or CPS courses to an existing program curriculum for students to gain the knowledge necessary to build reliable, efficient, and safe CPS or IoT systems.

It is time for a new engineering discipline that adapts to the reasons why IoT and CPS are different than existing engineering disciplines. History has shown that new engineering disciplines follow the newest technologies, and IoT and CPS are the newest technology trends. Electrical engineering emerged in the late 19th century with the invention of the electric motor. Chemical engineering emerged during the Industrial Revolution with the mass production of chemicals. Biomedical engineering rolled out in the early 1980s. Even the latest engineering discipline—software engineering—emerged as a result of the increased complexity of software systems. Now, with the capability of "things" that collect, aggregate, calculate, and send mounds of data for actuation, we argue that it is time for a new engineering/computer science discipline to emerge that is focused on this space.

 2469-7087/18/$33.00 © 2018 IEEE

A college-level program to educate a new workforce with the necessary skills to build effective and safe IoT and CPS systems is warranted. We suggest developing CPS and IoT engineering programs at colleges and universities that have established engineering departments, given the estimate of needing hundreds of thousands of IoT-educated engineers in the near future.[2] This does not suggest that vocational schools and other educational institutions cannot also help build this needed workforce—all help is needed.

A search on Indeed.com for US-based jobs that mentioned IoT resulted in more than 1,900 job opportunities. This doesn't include open positions in data analytics testing, algorithms, machine learning, or security, which are important disciplines in the design and implementation of CPS/IoT. In fact, the Bureau of Labor and Statistics predicts a 30 percent increase in jobs related to those technical domains by 2026.

IOT/CPS TRAINING

Academic institutions might already be considering an IoT-focused computer science degree or adapting curriculum from existing programs. To assist with that effort, co-author Voas, along with Phillip Laplante, mapped out five "Network of Things" (NoT) primitives that have been discussed by the National Institute of Standards and Technology (NIST)[4] relative to IEEE/ACM's 2013 computer science curricula knowledge areas (KAs; see Table 1).[5]

NoT is a term that applies to both CPS and IoT. The five primitives of all NoT systems include sensors (something that measures physical properties, such as RFID), aggregators (software to transform data from a sensor), a communication channel (data transmission, such as wired or wireless), an eUtility (software or hardware to execute processes, such as a database), and a decision trigger (which creates the final result, such as an actuator). Note that any specifically purposed NoT might not include all five. For example, some NoTs don't have sensors.

The easiest way to think about this is that the "things" are what make IoT unique. Many people question whether IoT is just marketing hype or if there is a science behind it. So, what is IoT? We'd better know before we start educating people about it.

IoT is an acronym of three letters. "I" (Internet) existed long before the acronym was termed and "o" does not matter, so "T" (things) is the letter in the acronym that we should pay attention to. So, the five NoT primitives define the "Lego-like" building blocks for any IoT-based system. The primitives are the "things," and this is where we need to focus our education efforts.

There are 18 KAs in computer science (for example, architecture and operating systems) that correspond well with understanding IoT in terms of the "things." Voas and Laplante recommended a set of topics to consider when creating new curricula or when modifying existing computer science curricula.[3] Further, if you are looking more at CPS issues than IoT concerns, modifying a systems engineering, electrical engineering, or mechanical engineering curricula might be worth pursing as well.

Table 1. IEEE/ACM computer science knowledge areas.[5]

1	Algorithms and complexity	10	Networking and communications
2	Architecture and organization	11	Operating systems
3	Computational science	12	Platform-based development
4	Discrete structures	13	Parallel and distributed computing
5	Graphics and visualization	14	Programing languages
6	Human-computer interaction	15	Software development fundamentals

7	Information assurance and security	16	Software engineering
8	Information management	17	Systems fundamentals
9	Intelligent systems	18	Social issues and professional practice

CPS/IOT PROGRAM STATUS

We reviewed CPS/IoT-related programs at the top 50 universities ranked by Collegechoice.net (an aggregate of *US News & World Report* and the National Center for Education Statistics) and TopUniversities.com (international universities) for IoT and CPS course offerings as of December 2017. More than half of those universities (see Table 2) had courses with a CPS/IoT focus, most of which were in graduate programs. Interestingly, more than half of those courses are taught in electrical engineering and computer engineering programs.

Table 2. Number of Internet of Things (IoT)/cyber-physical systems (CPS) courses at the top 50 ranked universities.

Universities with IoT/CPS courses	Total IoT/CPS courses	Undergrad courses	Graduate courses
28	49	17	32

The course descriptions of those 49 courses reveal that the IoT primitives are covered; however, only 11 percent seem to cover all five primitives (see Figure 1). These courses, "Interconnected Embedded Systems," "Networked Cyber-Physical Systems," "Internet of Things—Intelligent and Connected Systems," and "Body Sensor Networks in the Internet of Things," appear to be introductions to IoT and CPS technical and design understanding.

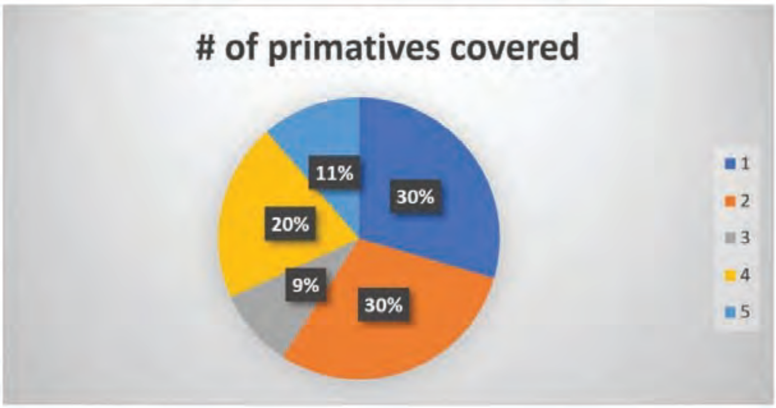


Figure 1. Number of IoT primitives covered in the courses at the top 50 ranked universities.

We reviewed more extensively some of the courses in CPS/IoT to gain a deeper understanding of the course content and structure. The courses reviewed focused on embedded systems with a CPS concentration, and on either CPS or IoT specifically. The difficulty in creating these courses is addressing the challenges of a CPS/IoT designer,[6] which include heterogeneous network technology integration, fault tolerance on the many individual devices in a system, prioritizing critical actions during system degradation situations, and distributed system energy management.

Despite these challenges, there are many open opportunities for course development. In addition to course content, there are numerous projects and use cases to include in these courses that would foster real-world experiences for students. For example, creating a “smart city” could involve installing sensor boxes around a city to monitor pedestrian or automobile flow to make informed decisions about traffic lights, bus stops, or where the next convenience store is located. A robot project might also be an effective learning tool by using camera and location sensors on the robot and then communicating the information (such as the location of the robot in the room) back to an embedded server. Another interesting project might be simulating a miniature factory by using a robotic arm and a controller to communicate to the outside world, where an app is developed to control the arm. These sample projects could be integrated into CPS or IoT courses.

Table 3 shows plausible examples of CPS or IoT course content. These examples are also mapped to the IEEE/ACM KAs discussed earlier (see Table 1). Considering that these could be CPS/IoT courses, they will also map to the five NoT primitives.

Table 3. Example CPS/IoT courses.

Course focus	Knowledge areas covered
Embedded systems with more focus on critical thinking about the effects of embedded software on the behavior, safety, and reliability of a CPS	1, 2, 3, 4, 10, 11, 12, 14, 15, 17
Implementation of functional prototype sensor/control networks (wired or wireless through available mobile device apps)	2, 3, 8, 9, 10, 12, 13, 16, 17
Learning to design embedded and CPS systems with real-time behaviors	1, 3, 4, 9, 10, 11, 12, 16
CPS applications focusing on resource management, timing constraints, distributed sensing, computation, control, modeling verification, and testing	1, 8, 9, 10, 11, 12, 13, 16, 17,
Embedded controls, field programmable gate array design, and server programming	1, 2, 8, 9, 10, 12, 14, 15, 16
CPS architecture and their vulnerabilities to cyber-attacks	1, 2, 3, 6, 7, 9, 10,12,14,17
IoT physical and logical architecture and functional blocks, communications protocols, smart objects, security, data analytics, system management, and ethical and environmental impact	1, 2, 3, 6, 7, 8, 9, 10, 12, 16, 17, 18
Focus on IoT by using RaaS (Robot as a Service) integrating a robot, sensors, and actuators into a cloud computing environment	1, 2, 3, 4, 5, 6, 9, 11, 12, 13, 14, 15, 16, 17
Focus on IoT to design and prototype an ambient intelligence system	2, 6, 9, 10, 12, 13, 14, 15, 16
A lab implementing the functionality of an entire facility to test specific concepts	2, 8, 9, 10, 11, 12

RECOMMENDATIONS

Elective courses are the least onerous way to begin the development of a CPS/IoT curricula. It is a challenge to revise an existing academic program and even more challenging to offer a new academic program—especially one that is to be accredited.

This article originally appeared in
IT Professional, *vol. 20, no. 4, 2018.*

In Table 3, we described examples of elective courses that could combine concepts from the major academic programs (for example, electrical engineering or computer science), as well as integrating CPS/IoT concepts into the course content. However, this is not a long-term solution given the workforce needs for employees with expertise in developing safe, reliable, and secure CPS/IoT systems. In other words, elective courses are a quick and easy way to pioneer a program and gauge interest among current and future students; however, for a long-term solution, academic institutions need to begin to define new curriculums and degrees. One option would be certificate programs that eventually grow into full undergraduate or graduate degree programs.

Another approach would be to integrate CPS/IoT concepts into existing program courses by developing learning modules. These modules would highlight specific CPS/IoT concepts. For example, researchers at Virginia Tech created CPS security-focused learning modules.[7] The easily accessible learning modules contain a specific learning objective with tools and hands-on exercises relevant for conventional embedded systems, control system design, and cybersecurity courses.

There is no doubt that a complete CPS/IoT curriculum will require core skills from multiple existing engineering and/or computer science programs. Courses on subjects such as embedded systems, computer security, software architecture, software construction, and others will apply. Accordingly, the path of least resistance is to create new programs by modifying existing programs, as many of these courses might already exist at the institution and only require slight modifications. This process can be used with other academic program KAs. This appears to be the most efficient way to create new CPS/IoT educational programs that are relevant and timely.

REFERENCES

1. L. Columbus, "Internet of Things Market to Reach $267B by 2020," *Forbes*, 29 January 2017; www.forbes.com/sites/louiscolumbus/2017/01/29/internet-of-things-market-to-reach-267b-by-2020/#f2ddc5609bd6.
2. S. Krishnamurthi, "More Education Needed to Realize the Internet of Everything," *Network World*, 10 December 2014; www.networkworld.com/article/2857890/internet-of-things/more-education-needed-to-realize-the-internet-of-everything.html.
3. J. Voas and P. Laplante, "Curriculum Considerations for the Internet of Things," *Computer*, vol. 50, no. 1, 2017, pp. 72–75.
4. J. Voas, *Networks of 'Things'*, report SP 800-185, NIST, 2016; doi.org/http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-183.pdf.
5. *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*, ACM and IEEE Computer Society, 2013; www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf.
6. P. Gonzalez-Nalda et al., "Building a CPS as an Educational Challenge," *Int'l J. Online Eng.*, vol. 10, no. 4, 2014, pp. 52–58.
7. P.P. Deshmukh, C.D. Patterson, and W.T. Baumann, "A Hands-on Modular Laboratory Environment to Foster Learning in Control System Security," *IEEE Frontiers in Education Conference* (FIE), 2016; doi.org/10.1109/FIE.2016.7757669.

ABOUT THE AUTHORS

Joanna F. DeFranco is an assistant professor of software engineering at Penn State University. Contact her at jfd104@psu.edu.

Mohamad Kassab is an associate research professor of software engineering at Penn State University. Contact him at muk36@psu.edu.

Jeffrey Voas is an IEEE Fellow. Contact him at j.voas@ieee.org.

Editor: Nayeem Islam ■ Qualcomm ■ nayeem.islam@gmail.com

Squeezing Deep Learning into Mobile and Embedded Devices

Nicholas D. Lane, *University College London and Nokia Bell Labs*
Sourav Bhattacharya and Akhil Mathur, *Nokia Bell Labs*
Petko Georgiev, *Google DeepMind*
Claudio Forlivesi and Fahim Kawsar, *Nokia Bell Labs*

In a relatively short time, deep learning principles and algorithms have transformed how the world processes, models, and interprets data.[1] For discriminative learning tasks routinely integrated into mobile and embedded systems—such as recognizing spoken words, objects, and faces—deep networks have been the state of the art for many years. Looking ahead to future device-based applications of learning, deep models are proving pivotal in the development of control algorithms for autonomous cars and drones (for example, for deep reinforcement learning). Deep models are also expanding into the area of core system issues—improving, for example, methods for encryption and compression.[2]

The blending of learning algorithms and mobile computing taking place today is only the beginning. We believe, in particular, that deep learning will play a prominent role in the evolution of smart devices (such as phones, watches, and embedded sensors) moving forward. It is therefore of paramount importance that we advance our understanding of how to simply and efficiently integrate current—and future—deep learning breakthroughs within constrained computing platforms (for more information, see the "Deep Learning under Constrained Devices" sidebar). This, along with continued research into the use of deep neural networks that support the diverse inference needs of sensor systems, will help produce radical improvements in how on-device context modeling and activity recognition is performed.

The emergence of mobile and embedded forms of deep learning has been slowed by the extreme resource overhead that it can easily introduce. Deep networks often contain hundreds of layers of interconnected nodes, and performing a single classification from a frame of sensor data can require computations over potentially hundreds of millions of parameters. Model representations and inference algorithms originally conceived for deep networks can easily overwhelm the resources of constrained platforms. In response to this resource barrier, the past 18 months have seen a surge in the investigation of resource-efficient deep learning for mobile and embedded platforms.

Promising early results are appearing across many domains, including hardware,[3,4] systems,[5,6] and learning algorithms.[7,8] Likely to further accelerate progress is the rate at which existing commercially supported deep learning tools, libraries, and frameworks have begun to address the specific needs of constrained devices (examples include TensorFlow, Caffe2, SNPE, Compute Library from Google, Facebook, Qualcomm, and ARM). These tools are starting to offer building blocks that enable fundamental research in this area by simplifying key steps such as runtime support on Android devices, processor-optimized low/mix precision matrix multiplication, or access to often unavailable heterogeneous device processors such as digital signal processors (DSPs) or GPUs.

In this short article, we provide an overview of the progress we have made toward overcoming a variety of core challenges facing deep learning for mobile and embedded devices, while also attempting to connect our findings to those of the wider community in the area. This discussion is largely focused on improvements seen within on-device execution of deep networks, which assumes the models are trained off-device. This is because execution (that is, inference) is the critical first step toward deep learning support, and it's the focus of almost all existing work, although exploration of on-device training has begun. Finally, given space constraints, we only superficially touch upon the ways in which deep learning is changing the face of activity and context recognition,[9] again limiting our focus to on-device examples.

 Published by the IEEE Computer Society 2469-7087/18/$33.00

The deep learning revolution has been powered by major advances in training algorithms, leaps in the availability of computing resources (primarily GPUs), and of course increased access to large-scale data. But at the core of any on-device, use of deep learning remains a neural architecture that must be efficiently executed.

PRIMER ON DEEP LEARNING INFERENCE AND ARCHITECTURES

Although a variety of deep model architectures have been developed, here we briefly describe two popular networks (shown in Figure A): deep neural networks (DNNs) and convolutional neural networks (CNNs). The role of training algorithms is to set the parameters of these neural architectures based on available data. This process is almost always assumed to occur off-device, and so the device itself is concerned with efficient inference.

Under a DNN, inference follows a feed-forward approach that operates on input data segments in isolation. The algorithm starts at the input layer and moves layer-wise sequentially while updating the activation states of all nodes within each layer. The process finishes at the output layer when all nodes in the layer have been updated. Finally, the inferred class is identified as the class corresponding to the output layer node with the greatest activation value. DNNs are often used in familiar mobile sensing tasks, such as spoken keyword spotting or identifying a speaker, but they're also use in extracting high-level human behaviors and contexts from inertial, location,[1] and (again) audio sensors.

Primarily used for vision and image-related tasks, CNNs are an alternative formulation of deep learning models. A CNN model contains one or more convolutional layers, pooling or subsampling layers, and fully connected layers. The objective of these layers is to extract simple representations from the input data and convert the representations into a more complex representation at much coarser resolutions within the subsequent layers. Lastly, fully connected layers often are used to help a CNN make predictions. CNNs can recognize a place type (such as a kitchen), accurately estimate age and gender, or more broadly recognize daily events from even noisy complex images, even those from wearable cameras.[2] Certain designs of CNN architectures like AlexNet or VGG[3] can be specialized to support many distinct tasks, and so their particular performance on constrained devices can become particularly important.

SYSTEM RESOURCE BOTTLENECKS

Model training is not the only computationally challenging process in deep learning. Even executing the straightforward inferencing step using a parameter-heavy model on a resource-limited device must overcome several challenges, including limited memory, limited computational power, and an unusually large inference time.[4,5] For example, deep models often have millions of parameters, and their storage on limited memory devices quickly becomes infeasible. Under low memory conditions, neural networks are often represented with low-precision parameters (8-bit or 16-bit) or by quantizing the weights of the architecture. Remarkably, even when heavily compressed with such methods, deep architectures can retain much of their accuracy. However, due to runtime memory limits, performing inference might still require frequent paging operations.

Inference time is also impacted by the overall number of computations. The availability of multiple cores and low-power processors on mobile platforms can be used to parallelize partial state updates of nodes to improve the inference time. Moreover, inferences often come with real-time requirements. Local execution of the memory- and computation-optimized models can potentially meet the requirements, overcoming intermittent connectivity problems prevalent in cloud-based systems.

Also, when running deep models continuously on embedded or wearable devices, high energy efficiency is crucial for maintaining a prolonged battery life. The energy consumption, among many things, mainly depends on the amount of computations, the use of low-power processors—such as digital signal processors (DSPs)—and the number of cache accesses. Thus, energy optimization requires a detailed understanding of the deep-model-execution pipeline on heterogeneous hardware platforms.

REFERENCES

1. J. Zhang et al. "DNN-Based Prediction Model for Spatio-Temporal Data," *Proc. 24th ACM SIGSPATIAL Int'l Conf. Advances in Geographic Information Systems* (GIS), 2016, article no. 92.
2. D. Castro et al., "Predicting Daily Activities from Egocentric Images Using Deep Learning," *Proc. 2015 ACM Int'l Symp. Wearable Computers*, 2015, pp. 75–82.
3. K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *Proc. Int'l Conf. Learning Representations* (ICLR), 2015; https://arxiv.org/pdf/1409.1556.pdf.
4. N.D. Lane et al., "An Early Resource Characterization of Deep Learning on Wearables, Smartphones, and Internet-of-Things Devices," *Proc. 2015 Int'l Workshop on Internet of Things towards Applications* (IoT-App), 2015, pp. 7–12.
5. J. Albericio et al., "Cnvlutin: Ineffectual-Neuron-Free Deep Neural Network Computing," *Proc. 43rd Int'l Symp. Computer Architecture* (ISCA), 2016, pp. 1–13; https://doi.org/10.1109/ISCA.2016.11.

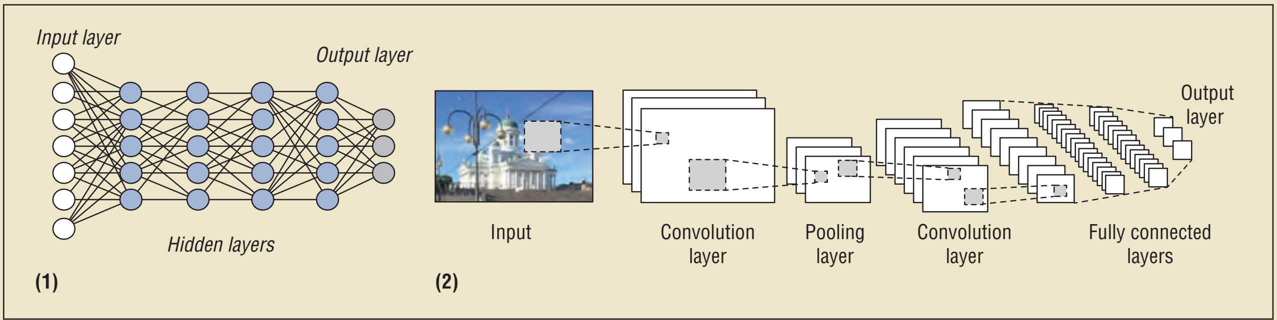


Figure A. Two popular neural network architectures: (1) deep neural networks (DNNs) and (2) convolutional neural networks (CNNs).

SMARTPHONES

EARLY SMARTPHONE SENSING RESULTS

In late 2014, we began our exploration into deep learning, starting with smartphones. These early investigations were motivated by two questions. First, could typical mobile and embedded sensing tasks, such as activity recognition and context sensing, be improved by the same deep learning approaches that were revolutionizing so many other inference domains? Second, how feasible was it to use these notoriously resource-heavy modeling techniques for user devices such as smartphones?

Fast forward to today, and deep networks for activity recognition—and smartphone sensing in general—have become much more mainstream. Researchers are developing powerful methods to train various deep architectures, raising the level of accuracy for models of human behavior.[9] Similarly, the ability to push neural networks into phone DSPs for low-power operation, a core innovation in our 2014 work (discussed next),[10] is an upcoming feature of Google's TensorFlow in partnership with Qualcomm.[11]

Deep Networks for Activity Recognition and Audio Sensing

We devised early deep learning solutions for well-known smartphone recognition tasks to quantify the benefits for on-device sensing.[10,12] A unique aspect of our approach was our focus on building constrained deep networks suitable for mobile and embedded devices. We wanted to know if deep learning was a viable and transformative replacement for the existing classifiers of mobile context and activities, grounded in shallow learning techniques. A core finding of our work was that for a range of sensing tasks, generic (nontask specific) deep networks could outperform state-of-the-art hand-selected features and shallow models—even when the deep networks were constrained to a size that made them more resource efficient than shallow alternatives.[10]

We then applied these findings to the audio domain and developed DeepEar,[12] a system for training and executing small-footprint deep neural networks (DNNs)—specifically, Restricted Boltzmann Machines (RBMs)—which were able to classify many audio contexts despite being a modest size of 2.3 million parameters each. As Table 1 summarizes, we stress-tested DeepEar, as well as a range of task-specific mobile audio classifiers, and on average, the accuracy was more than 30 percent higher for each task using DeepEar, even though each DNN was designed to execute not only within the CPU but even in the phone's DSP, a critical factor we explain next.

Low-Power Deep Networks via Heterogeneous Compute

Just as GPUs are a primary enabler for scaling up the training of larger and larger deep networks, we have found that non-CPU heterogeneous processors (such as DSPs) play a key role in scaling down deep networks for constrained devices. The DSPs in phones, for example, are sufficiently energy-efficient to compute on sensor data almost continuously while still supporting a device battery life beyond 24 hours.

Motivated by such efficiencies, we executed our proposed activity and audio targeting deep networks within the constraints of phone DSPs of the time—in

TABLE 1

A comparison of accuracy between our low-resource generic-task deep classifiers and existing hand-designed and task-specific (shallow) classifiers from the literature for various mobile sensing tasks. Note, reported microphone accuracy is lower than might be expected (for example, speaker identification), because experiments were conducted under severe acoustic conditions. (Experimental setup and classifier specifications appear elsewhere.[12,13] For each shallow classifier, we indicate the original venue of publication.)

Device type	Sensor	Sensing task	Task-specific shallow classifier (%)	Generic-task deep classifier (%)
Smartphone	Microphone	Ambient scene detection	81 (baseline from MobiSys 2009)	86
Smartphone	Microphone	Stress detection	62 (UbiComp 2012)	82
Smartphone	Microphone	Emotion recognition	72 (UbiComp 2010)	81
Smartphone	Microphone	Speaker identification	36 (Pervasive 2011)	57
Smartwatch	Accelerometer, gyroscope	Gesture recognition	68 (Activity Recognition in Pervasive Intelligent Environments 2010)	72
Smartwatch	Accelerometer, gyroscope	Physical activity recognition	82 (SenSys 2010)	93
Smartwatch	Light sensor, magnetic sensor, microphone, temperature sensor, proximity sensor	Location detection (indoor/outdoor)	87 (SenSys 2014)	94

particular, within memory footprints of just 8 Mbytes (using the Hexagon DSP of the Qualcomm Snapdragon 800).[10,12] DeepEar, under the Hexagon DSP, could run for 24 hours while using just 6 percent of a typical phone battery life with interleaved DNNs supporting four different audio tasks. In our follow-up system, DeepX,[5] we showed that by dividing models across a wide range of commodity phone processors (CPUs, DSPs, and GPUs), such efficiency gains were possible for not just small-scale DNNs but also other architectures, including even large image-based deep networks (such as the CNN AlexNet with 61 million parameters).

Our algorithms in DeepX allowed neural networks to be partitioned across different processor types within a local device using a runtime form of model compression that used singular value decomposition (SVD) to cope with processor constraints and minimize inter-processor overhead. Our smartphone prototype (on the Snapdragon 800) showed that this let various well-known deep models execute with efficiencies far in excess of baselines based on single processors or model compression alone (our prototype was up to seven times more energy efficient, for approximately a five percent loss in accuracy).

VGG AND MORE ON A SMARTWATCH

As techniques for deep learning on phones have matured, we have started studying how these issues manifest under smartwatches. The capabilities (compute and memory) of watches, coarsely speaking, lag phones often by one or two device generations; a typical Android smartwatch has not only 512 Mbytes of RAM and a multicore CPU but also a GPU and DSP. Watches are also natural for performing continuous and diverse behavior and context inferences—unlike phones, which can spend most of the day in pockets and bags. These two factors make it both conceivable and warranted for watches to join phones in performing nontrivial deep learning.

Transforming Watches from Smart to Deep

As in DeepEar,[12] our first proposed watch,[13] deep learning models were applicable to a range of common watch sensing tasks (shown in Table 1). Just as the DeepEar experiments had done for the smartphone audio domain, we demonstrated that typical inertial and wearable sensor data (such as accelerometer, barometer, and magnetometer data), fed into DNNs suitable in size for watches (around 200,000 parameters), could outperform existing task-agnostic classifiers from the literature.

This result further added to the understanding of feature representation learning by showing that these DNN models, produced by a single (off-watch) training framework, could outperform custom per-task combinations of hand-selected features and shallow models. On average, tasks were more than 7 percent more accurate compared to the best performing manually constructed classifier, while exerting a reasonable overhead.[13] For example, a commodity LG smartwatch could run one such RBM at 3 Hz and still maintain a 32-hour battery life.

Leveraging Layer Separation and Compression

Most examples of deep models—designed to process images, for example—dwarf the DNNs just described. The well-known VGG architecture can perform object recognition (and many other visual tasks) but at a cost of 138 million parameters or more. To prove the potential of smartwatches to support such demanding deep models, we showed that the VGG can be run locally on commodity smartwatches with a loss of approximately 3 percent accuracy (a tuneable parameter).[7] This was achieved primarily through a method applicable to any CNN, which reduces the computational bottleneck of applying thousands of convolutional kernels through what we call kernel separation. This technique replaces the 2D kernels defined during training with a pair of 1D vertical and horizontal kernels that, when used together, produce a result that approximates that of the original 2D version.

We coupled this optimization with the earlier described SVD-based model-compression technique for the fully connected layers at the end of the CNN, which simplifies the description of how nodes are connected and allows a further reduction in the number of parameters. We studied this approach on commodity watches under a variety of deep models, with VGG being the most resource intensive.[7] VGG, for example, executes in just under 1.2 seconds (a 2.7 times gain over conventional implementations) on LG smartwatches. These results, under some of the heaviest examples of deep models, pair with our low-resource DNN-based findings to show how deep solutions can not only improve over shallow methods but also be adopted in watches.

OVERCOMING SEVERE EMBEDDED CONSTRAINTS

As we have discussed, resource constraints present nontrivial barriers for deep learning on phones and watches. However, within embedded processors, these issues are magnified to extreme levels. Smartphones can address multiple Gbytes of RAM, but embedded processors, such as the ARM Cortex series, typically are limited to just hundreds or even tens of Kbytes. Similar resource differentials also extend into energy and compute domains. For these reasons, unlike the proliferation of phone-based deep learning in the last 18 months, few examples of deep learning under embedded constraints currently exist.

Toward filling this void, promising results are being seen in the form of binary deep architectures that are composed solely of 1-bit weights[14] instead of 32-bit or 16-bit parameters. Such architectures offer incredibly small models and remove the need for expensive multiplication operations, but their ability to perform well with real-world problems is still an open question. Solutions more closely tied to hardware will also undoubtedly play a key role in the area,

SMARTPHONES

such as the unique co-design opportunities for embedded-scale deep models that are built for field-programmable gate arrays processors,[3] or even emerging small form-factor deep learning accelerators (see, for example, https://uploads.movidius.com/1463156689-2016-04-29_VPU_ProductBrief.pdf).

Sparse Compression for Embedded Processors

Our contribution to the embedded area has been to devise a new form of model compression[7] that enables conventional DNNs to both fit and execute within the embedded processors, such as the ARM Cortex M3, and even the ARM Cortex M0! With this technique, fully connected layers of the DNN are represented using a sparse dictionary. As shown in Figure 1, dense matrices that capture the pair-wise dependencies of nodes (that is, weight matrices) are replaced with a code-book and sparse matrix that, together, closely approximate the dense original. We discovered a sparse-coding formulation that lets this approximation (and therefore the model accuracy) remain high. The dictionary is trained from the initial model representation, and a large saving in computation and memory results because nonzero elements can be ignored.

Compute savings under our approach are even further magnified, because, at execution, high-efficiency sparse matrix multiplication algorithms can be adopted in favor of conventional varieties that assume dense matrices. Although this method is only applicable to fully connected layers, it addresses the central embedded bottleneck of model size and still remains broadly useful, because the operations optimized are a key component to alternatives such as recurrent and convolutional architectures.

Experiences on the ARM Cortex

To measure the gains of our sparse-coding method for embedded processors, we tested DNNs for two audio tasks: speaker recognition and classification of the acoustic environments. We adopted an existing DNN architecture and training methods designed for low-resource platforms while still maximizing audio task robustness. Our findings showed, for example, that at the expense of 2 percent in accuracy, model compression by sparse coding can reduce these already optimized models by a factor of approximately 17 times for both tasks. In the case of speaker recognition, DNNs executed within our runtime that could leverage the sparsity of model representation showed a tenfold improvement in execution time within both ARM Cortex processors.

These gains make it feasible to run what are normally smartphone-class audio models in severely constrained processors. However, work remains to make deep models of this scale completely practical, because they still can't execute these models in real time—execution is still in the order of tens of seconds even to process a single five-second audio clip.

LOCAL EXECUTION OF MULTIPLE DEEP MODELS

Virtually all of the progress made thus far in mobile and embedded deep learning assumes that a single model executes on a constrained device. This is natural, because even a single deep model can present considerable technical challenges. However, most devices and applications will need to execute multiple models as part of their daily operations. For example, a wearable camera likely won't just recognize objects; it will also identify people and track facial expressions.

Between-model optimization opportunities exist most often when the collection of models perform related tasks (like image models), because each is trained independently, which lets natural redundancies emerge. For example, models that perform face recognition and object recognition will both learn layers that perform a type of edge detection during training, even though this operation could, in theory, be shared. Optimization opportunities such as this present an important class of performance improvements that has received little attention thus far.

Multiple Model Inference Pipeline

As a first step in addressing this issue, we designed an inference pipeline for wearables that targets the local execution of multiple image-based CNNs.[15] This

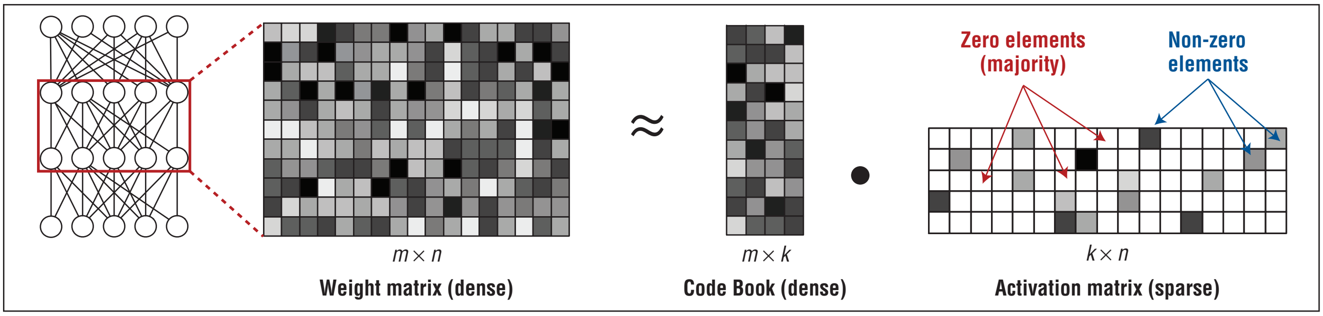


Figure 1. Illustration of our sparse-coding approach that factorizes dense matrices typically necessary to describe the connectivity between layers. A single dense matrix is approximated with two matrices; one is the weight code-book and the other is the sparse layer connectivity descriptor. We note a similar factorization is used in DeepX (not shown), but sparse coding is replaced by a light-weight singular value decomposition (SVD)-based method.

pipeline builds on a single fundamental optimization insight—namely, that CNNs are comprised of both *computation-heavy* convolutional layers and *memory-heavy* fully connected layers. Although convolutional layers only lightly tax the memory resources, they are computationally demanding. In contrast, fully connected layers place the exact opposite resource demands.

Due to these orthogonal resource demands of memory and compute, it's possible to schedule and batch layers together from multiple models to better maximize the resources of constrained devices and avoid bottlenecks that prevent multiple deep models from being executed. Our layer-centric execution framework for the inference stage of multiple CNNs focuses on optimal scheduling and batching decisions for device performance with a global view of all models, while still adhering to the layer dependencies of the neural network architecture.

Beyond this core idea, the execution framework incorporates memory caching of frequently used fully connected layers, selective use of SVD-based compression (described earlier), and logic that identifies the visual similarity in consecutive images to avoid unnecessary operations. Although designed for CNNs, the underlying concepts of this pipeline can generalize to other deep architectures.

DeepEye Wearable Camera

To study this multiple model pipeline, we integrated it within DeepEye—a prototype wearable camera based on a commodity processor (the Qualcomm Snapdragon 410) that offers execution of multiple CNN models without offloading computation to the cloud. DeepEye supports two use cases: lifelogging and vision assistance. Lifelogging seeks to log various everyday user experiences, with DeepEye realizing this through CNNs that can recognize objects, places, and faces and infer important image regions and how memorable an image is for the user. In contrast, vision assistance aims to help users who have low-vision capabilities by applying the same deep models that detect faces or objects, along with additional CNNs that infer age, gender, and emotions.

We compared the performance of DeepEye against the serial execution and single-model optimization alternatives. Experiments revealed that the latency for executing the multimodel inference pipeline is 10.10 seconds and 8.2 seconds for lifelogging and vision assistance, respectively (gains of 1.7 and 1.88 times over baselines, respectively).[15] These gains translate into a battery life of nearly 20 hours (1.4 times gain over the baseline), assuming images are captured every 30 seconds.

Deep learning on constrained devices, such as phones, watches, and even embedded sensors, is already well on its way to becoming mainstream. This is being enabled by a growing community of academic and industrial researchers who are bridging the worlds of machine learning, mobile systems, and hardware architecture.

Looking toward what is next, in the short term, we're likely to see continued leaps in activity and context-recognition accuracy, as insights from deep learning continue to propagate. We're also likely to see not just inference but also training being performed more routinely on devices. More fundamentally, applications of deep learning today are largely limited to classification tasks, yet the broader trend is for these algorithms to perform a wider range of computation. Within constrained devices, the potential definitely exists for them to begin to perform control and decision tasks, as well as more application logic, where their ability to learn and adapt dynamically to complex conditions might overcome some of the more brittle characteristics of sensory system behavior that have proven difficult to overcome. ■

SMARTPHONES

REFERENCES

1. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
2. G. Toderici et al., "Variable Rate Image Compression with Recurrent Neural Networks," *Proc. Int'l Conf. Learning Representations* (ICLR), 2016; https://arxiv.org/abs/1511.06085.
3. S. Han et al., "ESE: Efficient Speech Recognition Engine for Sparse LSTM on FPGA," *Proc. 2017 ACM/SIGDA Int'l Symp. Field-Programmable Gate Arrays* (FPGA), 2017, pp. 75–84.
4. R. LiKamWa et al., "RedEye: Analog ConvNet Image Sensor Architecture for Continuous Mobile Vision," *Proc. Int'l Conf. ACM/IEEE 43rd Ann. Int'l Symp. Computer Architecture* (ISCA), 2016, pp. 255–266.
5. N. Lane et al., "DeepX: A Software Accelerator for Low-Power Deep Learning Inference on Mobile Devices," *Proc. 15th Int'l Conf. Information Processing in Sensor Networks* (IPSN), 2016, article no. 23.
6. S. Han et al., "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding," *Proc. Int'l Conf. Learning Representations* (ICLR), 2016; https://arxiv.org/pdf/1510.00149.pdf.
7. S. Bhattacharya and N.D. Lane, "Sparsifying Deep Learning Layers for Constrained Resource Inference on Wearables," *Proc. 14th ACM Conf. Embedded Network Sensor Systems* (SenSys), 2016, pp. 176–189.
8. G. Huang et al., "Densely Connected Convolutional Networks," to appear in *Proc. 13th IEEE Conf. Computer Vision and Pattern Recognition* (CVPR), 2017.
9. N. Hammerla, S. Halloran, and T. Ploetz, "Deep, Convolutional, and Recurrent Models for Human Activity Recognition Using Wearables," *Proc. Int'l Joint Conf. Artificial Intelligence* (IJCAI), 2016; https://arxiv.org/abs/1604.08880.
10. N.D. Lane and P. Georgiev, "Can Deep Learning Revolutionize Mobile Sensing?" *Proc. 16th Int'l Workshop Mobile Computing Systems and Applications*, (HotMobile), 2015, pp. 117–122.
11. Qualcomm, "TensorFlow Machine Learning Now Optimized for the Snapdragon 835 and Hexagon 682 DSP," 9 Jan. 2017; www.qualcomm.com/news/snapdragon/2017/01/09/tensorflow-machine-learning-now-optimized-snapdragon-835-and-hexagon-682.
12. N.D. Lane, P. Georgiev, and L. Qendro, "DeepEar: Robust Smartphone Audio Sensing in Unconstrained Acoustic Environments Using Deep Learning," *Proc. 2015 ACM Int'l Joint Conf. Pervasive and Ubiquitous Computing* (UbiComp), 2015, pp. 283–294.
13. S. Bhattacharya and N.D. Lane, "From Smart to Deep: Robust Activity Recognition on Smartwatches Using Deep Learning," *Proc. Workshop on Sensing Systems and Applications Using Wrist Worn Smart Devices* (WristSense), 2016; http://ieeexplore.ieee.org/document/7457169.
14. B. McDanel, S. Teerapittayanon, and H.T. Kung, "Embedded Binarized Neural Networks," *Proc. Int'l Conf. Embedded Wireless Systems and Networks* (EWSN), 2017; www.eecs.harvard.edu/~htk/publication/2017-ewsn-mcdanel-teerapittayanon-kung.pdf.
15. A. Mathur et al., "DeepEye: Resource Efficient Local Execution of Multiple Deep Vision Models Using Wearable Commodity Hardware," to appear in *Proc. 15th Int'l Conf. Mobile Systems, Applications, and Services* (MobiSys), 2017.

Nicholas D. Lane is a senior lecturer (associate professor) at University College London and a principal scientist at Nokia Bell Labs. Contact him at niclane@acm.org.

Sourav Bhattacharya is a research scientist at Nokia Bell Labs. Contact him at sourav.bhattacharya@nokia-bell-labs.com.

Akhil Mathur is a research scientist at Nokia Bell Labs. Contact him at akhil.mathur@nokia-bell-labs.com.

Petko Georgiev is a research engineer at Google DeepMind. Contact him at pig20@cam.ac.uk.

Claudio Forlivesi is a research engineer at Nokia Bell Labs. Contact him at claudio.forlivesi@nokia-bell-labs.com.

Fahim Kawsar leads the Internet of Things research at Nokia Bell Labs. Contact him at fahim.kawsar@gmail.com.

This article originally appeared in
IEEE Pervasive Computing,
vol. 16, no. 3, 2017.

COLUMN: THINK PIECE

Critical Failure

Computer-Aided Instruction and the Fantasy of Information

Roderic N. Crooks
University of California, Irvine

Editor:
Bradley Fidler

The history of the use of various kinds of computers in education involves frequent triumphalist claims about the inevitable automation of instruction and equally frequent declarations of the failure of this project. This article situates both types of claims within a broader cultural understanding, one that holds that the human world is fundamentally informational and therefore amenable to improvement by computers.

In the pages of this magazine, Joy Rankin called for a history of social computing, encouraging historical scholarship to examine not only the successive engineering feats that produced modern computers but also "the activity of computing as a social and cultural phenomenon."[1] Central to this pursuit would be "an important but little studied area," the history of education and computers. This project might include records of educational computing projects (such as PLATO, the computer system that Rankin wrote about), software and logs created by students across various sites, and exploration of the range of conflicting meanings ascribed to computers and computing.[1] In what follows, I take up the last part of this challenge by asking about a central component of the meaning of social computing: the persistent vison of radically improving education through various forms of "computer-aided instruction." Just as the promise that computers will radically change education for the better has persisted for decades, so too have constant, bitter reports of the failure of computers to make good on this promise. The history of the use of computers in education is a story that oscillates between triumphant declarations of the arrival of an imminent future and equally vehement announcements of present failure.

Just as the promise that computers will radically change education for the better has persisted for decades, so too have constant, bitter reports of the failure of computers to make good on this promise.

Experts and hucksters alike have attempted to replace the teacher, the book, and the chalkboard as the primary media of education for much of the 20th century.[2] Early adopters of radio, then motion pictures, then television all claimed to be on the cusp of reforming instruction based on a new kind of educational media. These attempted transformations never quite happened, and these emergent media gradually

 Published by the IEEE Computer Society

found their places as *supplements* to human-led, bibliocentric, face-to-face instruction. For over half a century, computers in various configurations have continued to promise a change in instruction that never happens (the year of the Massive Open Online Course was six years ago, if anyone cares to remember[3]). According to one popular explanation, the tenacity of the vision of computer instruction is a feat of marketing. In this view, computers traveled from their industrial contexts into schools and homes via successive waves of engineering advances, ever-cheaper and more powerful products, and canny manipulation of consumer sentiment.[4] In this telling, technology goes where advertisers direct. That consumer culture trades in affect and sentimental relationships to products and brands cannot be denied.[5] But decades before there were any products to be marketed to the general public, computers were already educational. In a 1965 address to the American Federation of Information Processing Societies (predecessor of the IEEE), Ralph W. Gerard, dean of the Graduate Divisions of the newly formed University of California, Irvine, described a "tremendous opportunity for the future" predicated on the unison of the human mind and the useful but limited capacities of computers:[6]

> *What we are really facing, of course, is a symbiosis of both, combining the attributes of great speed and vast memory of the idiots that we call computer systems with the imaginative, creative, idiosyncratic, pattern-forming capacities of the human brain and mind.*

As Gerard saw it, individualized curriculum delivered via timesharing terminals would save money and allow university education to scale up to meet the demands of a growing population. Gerard's talk takes up key ideas in a longer history of 20th-century speculative technological projects, a heritage that connects Vannevar Bush's vision for the memex, an educational machine that could retrieve scholarly knowledge automatically;[7] Paul Otlet and Henri la Fontaine's establishment of the Mundaneaum, an archive and index of all the world's knowledge to be accessible via telephone and telegram;[8] and H.G. Wells' World Brain, a "Permanent World Encyclopedia" to be printed on microfilm.[9] It is from the lineage of these fantastic machines that computers came to be viewed as useful idiots ready to automatically (and cost effectively) take up labor-intensive forms of library work, effortlessly precise information retrieval, and automatic education.

Experts and hucksters alike have attempted to replace the teacher, the book, and the chalkboard as the primary media of education for much of the 20th century.

Before personal computers were marketed successfully to the general public, microcomputers of various kinds had already been placed in affluent public schools. Bill Gates learned to program in BASIC on a machine at his high school.[10] Steve Jobs and Stephen Wozniak learned about electronics and started building computers in the schools of Cupertino, California.[10] In 1976, before it had much of anything to sell, Apple Computer, Inc.'s Corporate Objectives stated:[11]

> *We also feel that Apple can contribute in certain special ways due to the unique nature of our products; i.e., improving the educational process through the use of small computers.*

By 1980, these "special ways" had already become ingrained in popular perceptions of computers. Apple famously marketed its products to public schools, but this strategy could not have succeeded without citing an extant understanding of what computers could do and be.[12] Journalism, social science, film, television, and advertising train the public to interpret consumer technology when its associations and valences are in flux.[13] These media tapped into existing fantasies (and anxieties) about computing and simultaneously stabilized a horizon of intelligibility around an exoteric object. Early advertising for personal computers positioned these newly available machines as both the means to accomplish white-collar affluence and the tool by which work and leisure would be effected. The personal computer then became both symbol and medium of the information age, a clever compression of the worlds of knowledge, leisure, work, play, research, and finance into the space of information.[14] An all-purpose machine for an all-purpose concept.

A stark counter-discourse of failure animates the history of computer-aided instruction in both popular and academic accounts. In a matter-of-fact dismissal of technological solutionism written in 1995, Alfred Bork—physicist, computer scientist, and foundational figure in the design of interactive, educational multimedia—asked, "Why Has the Computer Failed in Schools and Universities?"[15] Building on decades of research and teaching, Bork leveled a stern indictment of the state of the art of computer-mediated pedagogy by insisting, "We could have rebuilt education with technology many years ago."[15] Bork's complaint identified several culprits, including an emphasis on hardware rather than on learning or students, "elitist" software designed exclusively for expert users, and the pointless pursuit of innovation. Bork's broadside placed blame not on machines themselves, but on humans who refused to take advantage of the beneficial aspects of computers, their power, their organization, their logic. In this version of the story, it is humans who have become idiots by missing out on an essential quality of computers that could transform teaching and learning for the better, if only humans would get with the program.

Bork's fiery editorial evokes a feeling of world-weariness, a sense of fatigue at being forced to point out what is so indisputably self-evident. If the accomplishment of this self-evident link between computers and their value in instruction predates the birth of the personal computer, it has also survived its demise. For several years, I studied an attempt to use successors to the personal computer in urban education. During a period in 2013 to 2015, many schools in Southern California that serve minoritized communities, primarily Black and Latino, set out to provide every teacher, student, and administrator with a tablet computer to "close the digital divide and level the playing field, not only with educational access but technological access."[16] What is most striking about these projects is how insistently they call back to Apple's "special ways" of making education better, and also to Bork's insistence that machines, if let to do their work, could correct troubled schools. What these attempts to introduce a certain kind of computing in the poorest schools of Los Angeles produced was spectacular failure, one imputed not to tablet computers, but to the schools who refused to use them correctly.[17] This points to the importance of the ever-changing material forms of information and media technology, a way they have of keeping themselves new.[18] But the persistence of the commonsense assertion that computers in any form can reinvent instruction also points to a willful, collective suspension of disbelief.

Public failure of a technological project is a moment ripe for analysis, a way "to think of the social in terms of unfinished stories."[19] In the case of computers applied to education, failure has a way of reinforcing a story about how improvements in computer technology turn into improvements in society, despite all evidence to the contrary. At stake in the always unfinished project started by Gerard, Gates, Jobs, Wozniack, and others and kept alive by Bork and *Wired* magazine is not just the promotion of any particular regime of computing in instruction (since many kinds of computers and devices are already collecting dust or waiting for repair in any given school), but the belief that the world can be organized and improved by computing. The "special ways" that computing can improve work, school, government, and play are norms built on a decidedly narrow conception of information, a progressive and optimistic vision that takes the natural sciences as an exemplar for all forms of knowledge transmission and communication. Computers and education are only "made for each other" (as Bork put it) in a cosmology where the world is made of information, that banal and mysterious ether that is both fuel and precipitate of global capitalism. The trope of failure is, in short, symptomatic of a powerful fantasy that subsumes all actual and potential human knowledge into flows of machine-readable information. In such a vison, it is only a matter of time before all learning and thought become informational, despite the constant refusal of the world to conform to this structure.[20] If we recognize that very little of life resembles information processing, the social history of computing might suggest that it is the foundational myth of information that has failed rather than humans or machines alone.

REFERENCES

1. J Rankin, “Toward a History of Social Computing: Children, Classrooms, Campuses, and Communities,” *IEEE Annals of the History of Computing*, vol. 36, no. 2, April 2014, pp. 85–87.
2. R A Reiser, “A History of Instructional Design and Technology: Part I: A History of Instructional Media,” *Educational Technology Research and Development*, vol. 49, no. 1, March 2001, pp. 53–64.
3. L Pappano, “The Year of the MOOC,” *The New York Times*, 2 November 2012; https://nyti.ms/2kBM8OV.
4. Z Sofia, “The Mythic Machine: Gendered Irrationalities and Computer Culture,” *Education, Technology, Power: Educational Computing as a Social Practice*, Hank Bromley and Michael W. Apple, State University of New York Press, 1998.
5. E Illouz, “Emotions, Imagination and Consumption: A New Research Agenda,” *Journal of Consumer Culture*, vol. 9, no. 3, 2009, pp. 337–413.
6. R W Gerard, *Computers and Education*, ACM Press, 1965.
7. V Bush, “As We May Think,” *The Atlantic*, vol. 176, no. 1, July 1945.
8. W B Rayward, “Visions of Xanadu: Paul Otlet (1868-1944) and Hypertext,” *Journal of the American Society for Information Science*, vol. 45, no. 4, 1994, pp. 235–51.
9. D Muddiman, “The Universal Library as Modern Utopia: The Information Society of H. G. Wells,” *Library History*, vol. 14, no. 2, 1998, pp. 85–101.
10. M Campbell-Kelly, W Aspray, and N Ensmenger, *Computer: A History of the Information Machine*, Westview Press, 2014.
11. “Corporate Objectivies,” *Apple Computer, Inc. Records, 1977-1998. Series M10007. Box 5. Folder 20.*, Apple Computer, Inc., 1976.
12. A Watters, “How Steve Jobs Brought the Apple II to the Classroom,” *Hack Education*, 25 February 2015; http://hackeducation.com/2015/02/25/kids-cant-wait-apple.
13. M Z Newman, *Atari Age: The Emergence of Video Games in America*, MIT Press, 2017.
14. R E Day, *The Modern Invention of Information: Discourse, History, and Power*, SIU Press, 2001.
15. A Bork, “Guest Editorial: Why Has the Computer Failed in Schools and Universities,” *Journal of Science Education and Technology*, vol. 4, no. 2, 1995, pp. 97–102.
16. *Common Core Technology Project*, Los Angeles Unified School District, 2013; http://achieve.lausd.net/Page/627.
17. I Lapowsky, “What Schools Must Learn from LA’s iPad Debacle,” *Wired*, 8 May 2015; https://www.wired.com/2015/05/los-angeles-edtech.
18. W H K Chun, *Updating to Remain the Same: Habitual New Media*, MIT Press, 2016.
19. V Das, *Life and Words: Violence and the Descent into the Ordinary*, University of California Press, 2007.
20. B Goertzel, “Human-Level Artificial General Intelligence and the Possibility of a Technological Singularity,” *Artificial Intelligence*, vol. 171, no. 18, December 2007, pp. 1161–1173.

ABOUT THE AUTHOR

Roderic N. Crooks is an assistant professor in the Department of Informatics at the University of California, Irvine. He studies information technology use in minoritized communities, frequently in the context of public schools. His recent work has appeared in *Science, Technology & Human Values*, *Education Media International*, and *The International Encyclopedia of Communication Theory and Philosophy*. Contact him at crooksr@uci.edu.

This article originally appeared in
IEEE Annals of the History of Computing, *vol. 40, no. 2, 2018.*

Editor: **Philippe Kruchten**
University of British Columbia
pbk@ece.ubc.ca

How Best to Teach Global Software Engineering?

Educators Are Divided

Sarah Beecham, Tony Clear, Daniela Damian, John Barr, John Noll, and Walt Scacchi

WITH GLOBAL SOFTWARE engineering (GSE) becoming standard practice, today's software engineering students will be tomorrow's global software engineers. So, the education systems underpinning the profession will need to change accordingly. However, current approaches to teaching software engineering are outdated and lack authenticity, as Florian Matthes and his colleagues noted:

> *When considering the personal requirement today's software engineers are facing in their daily work life, it is surprising to see that teaching GSE at universities is still in its infancy.*[1]

GSE is an established field, and nearly all practitioners and academics agree that graduating students must have experience in it. A report from the 20th Annual Conference on Innovation and Technology in Computer Science Education reviewed the GSE education literature, exposed the challenges to teaching GSE, and provided a framework for meeting these challenges in a university setting.[2]

To stimulate debate on how to change current approaches to teaching software engineering to reflect the global workplace, Sarah Beecham asked Tony Clear, Daniela Damian, John Barr, John Noll, and Walt Scacchi to discuss how they inject realism into their courses. (This Oxford-style debate took place at the GSE education workshop at UC Irvine in August 2016; for workshop details, visit gse.sivrex.com.) Although they all agreed that changes are necessary, their approaches differed considerably. Clear and Damian argued that the best way to emulate the workplace is to engage in cross-university, multisite courses. In contrast, Barr suggested that having students contribute to open source projects gives them real-world experience without the overhead involved in cross-university courses. Finally, Noll and Scacchi argued for using online simulations and games to provide students a range of experiences that wouldn't be possible within the constraints of a university term.

The following provides an overview of the approaches they discussed, in their own words.

The Multisite, Cross-University View (Clear and Damian)

In our work with students, we seek to conduct authentic global virtual collaborations in cross-university courses. Our key goals are to develop global collaborative capabilities, develop cross-cultural understanding, and demonstrate the challenges and complexities of working

 Published by the IEEE Computer Society

in global virtual teams, thereby fostering international understanding, peace, and global sustainability.

In doing so, we've also challenged ourselves to work with global colleagues in complex, sophisticated ventures. We've taken what has often been pioneering work, in which the course becomes a living laboratory, as an opportunity to engage students in research-based teaching. Through this teaching, we model and encourage inquiry-based learning. This form of teaching or learning isn't easy or comfortable, and mistakes and frustrations abound. It's true that we don't have all the answers, but why should we shield our students from that? This is how students develop the insight and skills to work effectively and sensitively as tomorrow's global practitioners.

Over two decades of diverse collaborations, we've inquired deeply into GSE and GSE education jointly with our students, created course models and instances, and generated new knowledge. We've developed global friendships with colleagues and students and seen these courses open doors for graduates. We argue that students learn best about GSE by doing GSE. Optimally, that occurs in a structured, conscious global learning experience—although at times we must eat some of our own dog food!

Can we really teach GSE competencies in the classroom? How do we expose students to the reality of complex working relationships specific to globally distributed software development? How do we mentor them through the sustained effort of finding strategies for successful global-software-development projects?

Not only can GSE learning be achieved in the classroom, but also a university course offers a safe environment for students to experiment and work through different strategies when facing new challenges. We believe that students can learn GSE competencies in the classroom but that this requires an educational environment that offers realistic challenges and more questions than answers. Our experience designing and evaluating GSE teaching frameworks in which students engaged in hands-on GSE projects indicates that these courses resembled, as much as possible, the reality of the software industry.

The outcome is convincing. For example, we recently tried out distributed-Scrum practices in a project led by a real client and involving distributed university environments. The students experienced realistic challenges of working with remote peers and the client across organizational, cultural, and temporal boundaries. In keeping with agile processes, the students engaged in ongoing reflection about their challenges, their response to these challenges, and GSE learning. The empirical evidence we collected on the students' learning of GSE competencies shows clearly that they learned, for example, to minimize cross-boundary communication when allocating work to respond to the client's feature requests.[3]

However, a note of caution: Despite this success, setting up, teaching, and evaluating student work in such educational environments requires more effort, strategy, and instructor resilience than in traditional courses. Enabling students' learning of global software development has many facets. However, multisite, multi-university courses let educators design an experience to best relate to the current processes, supporting tools, and realities of GSE. Despite the challenges we experienced running these courses, we should continue to do so.

> A university course offers a safe environment to experiment with GSE strategies.

A Pragmatic Approach to GSE Education (Barr)

Education in an academic setting must always be a facsimile of reality. In particular, any education must take place in an artificial framework—namely, the academic institution, which places constraints on the pedagogy. Challenges include these issues:

- *Course design and organization.* Universities and instructors, even within a country, have different philosophical beliefs about how to structure and assess courses.
- *Students.* Students from different universities and cultures have different work ethics, skill levels (grad versus undergrad), expectations from instructors, and expectations about class requirements.

- *Tools*. Universities, instructors, and students have experience with different communication and development tools. Sometimes, these tools are determined at the department or university level, which makes changing them difficult.
- *Class management*. Course instructors generally have set responsibilities (lectures, grading, office hours, and so on), which become more problematic when spread across multiple universities and geographic locations. Which instructor is responsible for what? How do instructors support students in remote locations? How are student teams managed across several locations? How does the instructor manage student relationships with remote clients?
- *Sustainability, scalability, and reusability*. How do you scale a GSE education class? Such scaling greatly magnifies the GSE challenges, and instructors normally receive few if any new resources to deal with them.

As a result of these intrinsic challenges, I propose an alternative to the multi-university approach: introduce students to GSE through participation in a global open source project. Open source projects have many advantages. If the project is indeed global, students are introduced to GSE challenges and can participate in the GSE process—GSE education's major goals. Open source projects use specific tools and processes that students will need to master; this provides essential experience in using communication and development tools. As students participate in development, they communicate globally with a team that's experienced in GSE and in integrating new developers.

This approach also ameliorates many of GSE education's constraints. The GSE infrastructure already exists, the development tools are set and typically are universally known and used (for example, git, Internet Relay Chat, wikis, and blogs), the languages are usually well known (for example, Python, Java, and C++), and many resources are available to new developers. An open source project is also highly motivating; it's a real project for a solution that's often in widespread use, and the open source community generally is accepting and encouraging. Because there's such a large selection of open source projects to choose from, instructors can choose a project (or a piece of a project) that fits their course objectives and their students' abilities.

Finally, many of the institutional challenges are vastly reduced. Instructors don't have to deal with differing philosophical approaches, manage remote students, negotiate schedules, or deal with many of the constraints of a multi-university approach.

Of course, an open source project isn't the same as a GSE project in a commercial company. However, it meets GSE education's general learning objectives and reduces the challenges significantly. I believe that, for most universities, it's the most cost-effective and practical approach.

> For most universities, open source projects are the most cost-effective and practical approach.

Why Not Use Simulations? (Noll and Scacchi)

Regardless of how much effort educators devote to making a distributed-project course realistic, it will always have constraints impeding its fidelity. University terms rarely last longer than 16 weeks, students have other classes and obligations competing for their time, and they don't earn a salary.

Nevertheless, a GSE course is still a simulation of a real industrial software development project. So why not go the full distance and train students using an online simulation or game? After all, airline pilots train extensively in flight simulators, which let them experience critical situations without putting themselves at risk. A "GSE simulator" could have the same advantage, letting students experience problems and make mistakes without putting their projects (and grades) at risk. A simulator can also compress time, simulating an entire project in an hour, letting students run many trials involving different project scenarios or strategies. And, simulations and games can be fun.

Educators can use simulations and games focusing on GSE processes in many interesting ways, providing educational and research affordances that are too difficult, costly, or lengthy to realize in live GSE course projects. Such affordances can include the capability to

- progressively guide students through GSE processes across difficulty levels, through short, self-paced training scenarios;
- monitor and incrementally assess student progress during simulated GSE process enactments;
- control the triggering or emergence of situated or contextual problems, conflicts, misalignments, time-zone asymmetries, and so on that can arise in GSE projects;
- capture and replay simulated GSE process enactments, allowing for close-up analysis and retrospective diagnosis of the actions taken and their consequences;
- incorporate live or simulated GSE tools and repositories;
- simulate benign to problematic GSE project circumstances such as mid-project staff termination and budget and schedule reductions; and
- accommodate student errors, mistakes, and process enactment failures as learning experiences that are safe, don't require remote confederates, and can simulate differences or gaps in cultural practices and diversity.

Simulations and games for GSE are no panacea; they're only as good as their developers have allowed. Poor design or implementation, or ill-conceived and poorly matched ontologies underlying the simulation, can render a simulator or game ineffective. But well-designed simulations and games offer capabilities that are scalable and open to experimentation and replicability in ways that are much easier to adopt and implement than with live role-playing GSE projects.

The approaches described here are, of course, simulations of the real workplace. The key message is that educators are changing how they teach, to try to reflect industry needs. Industry is calling for more and better-skilled software engineers; perhaps this is one way to address the skills shortage.

Airplane pilots train extensively in flight simulators. A "GSE simulator" could have the same advantage.

Acknowledgments

Science Foundation Ireland grant 13/RC/2094 partly supported this research.

References

1. F. Matthes et al., "Teaching Global Software Engineering and International Project Management—Experiences and Lessons Learned from Four Academic Projects," *Proc. 3rd Int'l Conf. Computer Supported Education* (CSEDU 11), 2011, pp. 5–15.
2. T. Clear et al, "Challenges and Recommendations for the Design and Conduct of Global Software Engineering Courses: A Systematic Review," *Proc. 2015 ITiCSE on Working Group Reports* (ITiCSE-WGR 15), 2015, pp. 1–39; http://dx.doi.org/10.1145/2858796.2858797.
3. M. Paasivaara et al., "Learning Global Agile Software Engineering Using Same-Site and Cross-Site Teams," *Proc. 37th Int'l Conf. Software Eng.* (ICSE 15), vol. 2, 2015, pp. 285–294.

SARAH BEECHAM is a senior research fellow at Lero—The Irish Software Research Centre. Contact her at sarah.beecham@lero.ie.

TONY CLEAR is an associate professor in the School of Engineering, Computer and Mathematical Sciences at the Faculty of Design & Creative Technologies at the Auckland University of Technology. Contact him at tony.clear@aut.ac.nz.

DANIELA DAMIAN is a professor of software engineering at the University of Victoria. Contact her at danielad@uvic.ca.

JOHN BARR is an associate professor in Ithaca College's Department of Computer Science. Contact him at barr@ithaca.edu.

JOHN NOLL is a research fellow at Lero—The Irish Software Research Centre. Contact him at john.noll@lero.ie.

WALT SCACCHI is a senior research scientist and member of the research faculty at the Institute for Software Research at the University of California, Irvine. Contact him at wscacchi@ics.uci.edu.

This article originally appeared in IEEE Software, *vol. 34, no. 1, 2017.*

TECHNOLOGY

Expedia, Inc.

has openings for the following positions in **Bellevue, WA**:

Managers, Engineering (Job ID#:728.2159): Manage the architecture, lead technical design, construction testing, and implementation. **Senior Data Engineers (Job ID#: 728.5377):** Drive data investigations across organizations and deliver resolution of technical, procedural, and operational issues to completion. **Managers, Software Development (Job ID#: 728.5487):** Provide technical leadership to local and remote developers. **Oracle Applications Developers (Job ID#: 728.4750):** Design, develop, implement, customize, and integrate Oracle e-Business Suite applications with other systems. **Data Scientists (Job ID#: 728.4428):** Create and deploy machine learning models that power a vast range of applications related to fighting fraud and mitigating risk. **BI Development Engineers (Job ID#: 728.5056):** Build robust, well-designed and well-engineered business intelligence development engineering solutions.

Expedia, Inc. has openings in **Chicago, IL** for:

Software Engineers (Job ID#: 728.5470) Design, implement, and debug software for computers including algorithms and data structures. May telecommute from home.

To apply, send resume to: Expedia Recruiting, 333 108th Avenue NE, Bellevue, WA 98004. Must reference Job ID#.

EAN.com LP

has openings in **Bellevue, WA**:

Security Engineers (Job ID#: 728.1840) Develop and consult on the security architecture and strategies for enterprise cyber security controls.

To apply, send resume to: EAN Recruiting, 333 108th Avenue NE, Bellevue, WA 98004. Must reference Job ID#.

CarRentals.com, Inc.

has openings in **Bellevue, WA**:

Engineering Managers (Job ID#: 728.2394) Lead team of developers in the design, implementation, and debugging of software for computers including algorithms and data structures.

To apply, send resume to: CarRentals Recruiting, 333 108th Avenue NE, Bellevue, WA 98004. Must reference Job ID#.