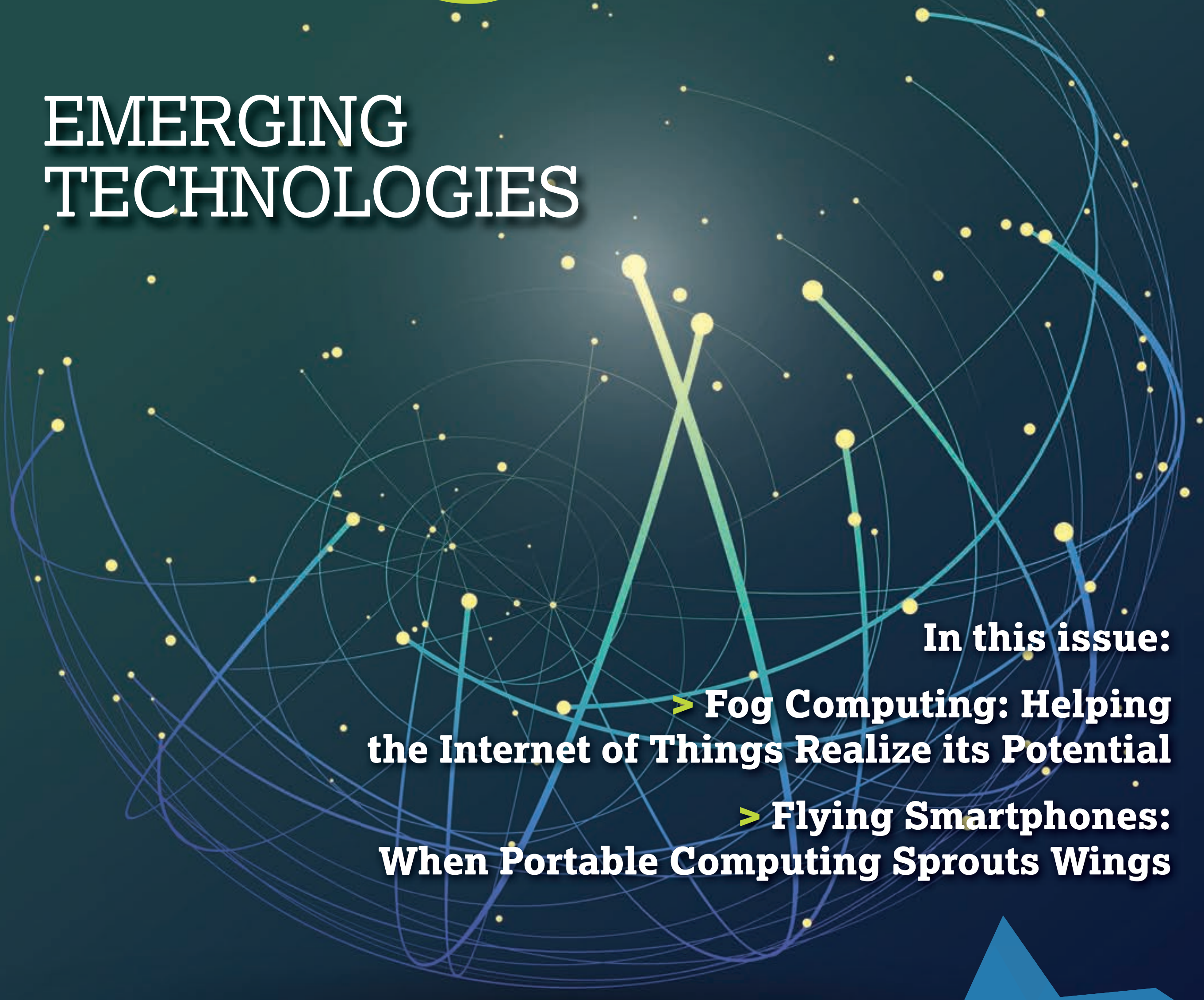
COMPUTING

edge

EMERGING TECHNOLOGIES



In this issue:

- **Fog Computing: Helping the Internet of Things Realize its Potential**
- **Flying Smartphones: When Portable Computing Sprouts Wings**

JANUARY 2017

www.computer.org

IEEE

IEEE computer society

COMPUTING edge

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

Printed with inks containing soy and/or vegetable oils

IEEE

IEEE computer society

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

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

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

JANUARY 2017 • VOLUME 3, NUMBER 1

COMPUTING edge

Departments

IEEE Computer Society Is Where You Choose the Resources that Fit Your Career

Find the membership that fits you best. IEEE Computer Society lets you choose your membership — and the benefits it provides — to meet your specific career needs. With four professional membership categories and one student package, you can select the precise industry resources, offered exclusively through the Computer Society, that will help you achieve your goals.

Select your membership	Preferred Plus		Training & Development		Research		Basic		Student
	$60 *IEEE Member*	**$126** *Affiliate Member*	**$55** *IEEE Member*	**$115** *Affiliate Member*	**$55** *IEEE Member*	**$115** *Affiliate Member*	**$40** *IEEE Member*	**$99** *Affiliate Member*	**$8** *Does not include IEEE membership*
Computer magazine (12 digital issues)*	✓		✓		✓		✓		✓
ComputingEdge magazine (12 issues)	✓		✓		✓		✓		✓
Members-only discounts on conferences and events	✓		✓		✓		✓		✓
Members-only webinars	✓		✓		✓		✓		✓
Unlimited access to *Computing Now*, computer.org, and the new mobile-ready myCS	✓		✓		✓		✓		✓
Local chapter membership	✓		✓		✓		✓		✓
Safari Books Online (600 titles and 50 training videos)	✓		✓						✓
Skillsoft online solutions (courses, certifications, practice exams, videos, mentoring)	✓		✓						✓
Two complimentary Computer Society magazine subscriptions	✓				✓				
myComputer mobile app	*30 tokens*				*30 tokens*				*30 tokens*
Computer Society Digital Library	*12 FREE downloads*		*Member pricing*		*12 FREE downloads*		*Member pricing*		*Included*
Training webinars	*3 FREE webinars*		*3 FREE webinars*		*Member pricing*		*Member pricing*		*Member pricing*
Priority registration to Computer Society events	✓								
Right to vote and hold office	✓		✓		✓		✓		
One-time 20% Computer Society online store discount	✓								

* *Print publications are available for an additional fee. See catalog for details.*

www.computer.org/membership

IEEE computer society

CS FOCUS

Magazine Roundup

The IEEE Computer Society's lineup of 13 peer-reviewed technical magazines covers cutting-edge topics ranging from software design and computer graphics to Internet computing and security, from scientific applications and machine intelligence to cloud migration and microchip manufacturing. Here are highlights from recent issues.

Computer

Computer's January 2017 special issue celebrates **the magazine's 50th anniversary**, looking back at its past and ahead at its future.

IEEE Software

Changes in the industry are making the software architect's job more challenging. The key drivers are the growing role of software in systems; the increasing emphasis on reuse, agility, and testability during development; and several quality-related factors affected more and more by architectural choices, according to "**The Changing Role of the Software Architect**," from *IEEE Software*'s November/December 2016 issue.

IEEE Internet Computing

JavaScript has already made inroads as an integrated technology stack for building user-facing applications. In "The Future of Big Data Is ... JavaScript?" which is in *IEEE Internet Computing*'s September/October 2016 issue, the authors explore the idea of **using JavaScript in big data platforms**.

Computing in Science & Engineering

"Risks," from *CiSE*'s November/December 2016 issue looks at how we address perceived and actual **risks in scientific programming** and how we can mitigate them from the technical and project-management perspectives.

IEEE Security & Privacy

Users aren't **the problem with security**, according to author Bruce Schneier in "Stop Trying to Fix the User," from *IEEE S&P*'s September/October 2016 issue. He says the problem is that we've designed our computer systems' security so badly that we force users do many counterintuitive things.

IEEE Cloud Computing

Cloud computing systems and services have become important cyberattack targets. To protect cloud platforms, infrastructures, hosted applications, and data storage, we must

Published by the IEEE Computer Society

address security challenges from multiple perspectives, including application outsourcing, anonymous communication, and secure multiparty computation, according to *IEEE Cloud Computing*'s September/October 2016 special issue on **cloud security**.

IEEE Computer Graphics and Applications

National-security agencies continue to push many advancements in computer graphics. *CG&A*'s November/December 2016 special issue explores the implementation of **computer graphics for defense applications**. The US Army and Navy have largely funded the work addressed in this issue's articles. Understanding the direction of military-related computer-graphics research can provide insight into future civilian uses, as well as provide a path to novel ideas.

IEEE Intelligent Systems

IEEE Intelligent Systems' November/December 2016 special issue on **innovative interaction** includes articles on the influence of rating prediction on group recommendation technology's accuracy, and a proposed architecture for a network of stationary sensors and mobile robots that provides distributed ambient intelligence.

IEEE MultiMedia

IEEE MultiMedia's October–December 2016 special issue on **advances in image, audio, and social multimedia computing** includes articles on nonparametric quality assessment of natural images, in-vehicle personalized audio zones, and a neural network for quality-of-experience estimation in mobile communications. There are also pieces on unsupervised speaker identification for TV news and expressive modulation of neutral visual speech.

IEEE Annals of the History of Computing

IEEE Annals' October–December 2016 issue is the second of two special issues on **security** based on the US National Science Foundation's Computer Security History Workshop. The workshop gathered historians and experts at the Charles Babbage Institute in July 2014 and advanced scholarship on many different critical aspects of computer-security history. This *IEEE Annals* issue includes articles on the early histories of security vendor Symantec, intrusion-detection expert systems, and edge cryptography and the codevelopment of computer networks and cybersecurity.

IEEE Pervasive Computing

Healthcare faces many privacy challenges, including patient expectations for confidentiality and sensors that collect nontraditional health-related data. These challenges make healthcare a unique environment for privacy. "Privacy Is Healthy," from *IEEE Pervasive Computing*'s October–December 2016 issue, discusses what ubiquitous-computing researchers and practitioners can do to improve **healthcare privacy**.

IT Professional

The digital technologies that underlie computers, robots, and smart equipment are rapidly evolving. They are becoming more powerful and are transforming organizations faster than in the past, changing the world of business and enabling organizations to achieve significant competitive advantages in the process. *IT Pro*'s November/December 2016 special issue on **digital innovation and transformation** examines this important trend and looks at future opportunities.

IEEE Micro

IEEE Micro's November/December 2016 special issue on the **Internet of Things (IoT)** includes articles on heterogeneous wireless sensor nodes that target the IoT, heterogeneous distributed shared memory for lightweight IoT devices, and the feasibility of attribute-based encryption on IoT devices. There are also articles on emergent behaviors in the IoT, the visual IoT, and work on a self-learning and energy neutral IoT.

Computing Now

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

Defining Data Clusters

Ali Jadbabaie, University of Pennsylvania

This installment highlighting the work published in IEEE Computer Society journals comes from IEEE Transactions on Network Science and Engineering.

Clustering has been extensively studied in data analysis. However, clustering theories have thus far been unsatisfactory to justify use of the many proposed algorithms. A fundamental question is this: How do we define a cluster in a set of data points?

In "A Mathematical Theory for Clustering in Metric Spaces," Cheng-Shang Chang and his colleagues attempt to answer this question by considering a set of data points associated with a distance measure, or metric (*IEEE Trans. Network Science and Eng.*, vol. 3, no. 1, 2016, pp. 2–16). They propose a new cohesion measure with regard to the distance measure. Using this cohesion measure, they define a cluster as a set of points cohesive to themselves, and provide various equivalent statements with intuitive explanations to support their definition.

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

The authors then consider a second question: How do we use this definition to find clusters and good partitions of clusters? Chang and his colleagues offer two algorithms: hierarchical agglomerative and partitional. Unlike standard hierarchical agglomerative algorithms, their algorithm stops specifically at partitions of clusters. Their partitional algorithm—dubbed the *K*-sets algorithm—is quite novel and interesting. Unlike the *K*-means algorithm (for data points in a Euclidean space) that assigns points to the closest centroid, the *K*-sets algorithm assigns points to the closest set with regard to the triangular distance—the average difference of the three sides of a triangle formed by the point and two randomly selected points in the set. As such, the *K*-sets algorithm performs well when a cluster can't be represented by a single point.

The authors also find that the duality between a distance measure and a cohesion measure leads to a dual *K*-sets algorithm for clustering a set of data points with a cohesion measure. The dual *K*-sets algorithm converges in the same way as a sequential version of the classical kernel *K*-means algorithm. The key difference is that a cohesion measure doesn't have to be positive semidefinite.

A video demonstration of the *K*-sets algorithm being used to detect six clusters on a plane from an initial random partition is available at imgur.com/b5KQo9I.

ALI JADBABAIE is an adjunct professor of electrical and systems engineering at the University of Pennsylvania, and director of MIT's Sociotechnical Systems Research Center. Contact him at jadbabai@seas.upenn.edu.

Published by the IEEE Computer Society

EDITOR'S NOTE

ComputingEdge Looks at Emerging Technologies

Computing is a fertile field for researchers, whose work frequently yields important new technologies. These innovative approaches create new possibilities and opportunities for users and frequently change the computing landscape. This *ComputingEdge* issue looks at some of today's important emerging technologies.

The US High-Assurance Cyber Military Systems project has conducted experiments with complex networked-vehicle software. According to *IEEE Software*'s "Requirements and Architectures for Secure Vehicles," attention to requirements and system architecture, along with verified approaches that remove known vulnerabilities, could yield vehicles secure from attacks.

Fog computing is designed to deal with the huge amounts of data that the Internet of Things generates but that traditional systems, the cloud, and even edge computing can't handle. This technology is discussed in *Computer*'s "Fog Computing: Helping the Internet of Things Realize its Potential."

Lightweight, highly autonomous, interactive drones are emerging as the next big thing in consumer electronic technology, according to "Flying Smartphones: When Portable Computing Sprouts Wings," from *IEEE Pervasive Computing*.

The authors of IEEE *MultiMedia*'s "Eye-Controlled Interfaces for Multimedia Interaction" present initial results from work with the gaze-based control paradigm that they developed. They are investigating how eye-based interaction techniques can be made precise and fast enough to let disabled people easily interact with multimedia information.

JavaScript has already made serious inroads as an integrated technology stack for building user-facing applications. *IEEE Internet Computing*'s "The Future of Big Data Is ... JavaScript?" explores the idea of using JavaScript in big-data platforms.

The healthcare industry's transformation to widespread digitization is yielding systems that promise to use ever-improving data-driven evidence to enable doctors to make more precise diagnoses and institutions to identify at risk patients for intervention. It will also allow clinicians to develop more personalized treatment plans and researchers to better understand medical outcomes within complex patient populations. Advanced data-visualization tools could play a critical role in this process. "Data-Driven Healthcare: Challenges and Opportunities for Interactive Visualization," from *IEEE Computer Graphics and Applications*, reviews healthcare-related visualization challenges.

In *IT Professional*'s "Data-Intensive Science," the author looks at data science applied specifically to scientific disciplines. The article highlights some of the policy, research, and infrastructure developments that are turning data-intensive science into a significant employment opportunity. It also mentions several disciplines in which data-intensive science has taken hold and discusses a key concern.

Editor: **Jane Cleland-Huang**
DePaul University
jhuang@cs.depaul.edu

Requirements and Architectures for Secure Vehicles

Michael W. Whalen, Darren Cofer, and Andrew Gacek

DO YOU TRUST the software in your vehicle? Recent exploits have let hackers remotely control aspects such as brakes and steering, perform surveillance and eavesdropping, and even remotely steal a car. It's speculated that Iran landed a US stealth drone at an Iranian airfield through a GPS spoofing attack. Recent research in self-driving cars and multivehicle coordination requires ever-more software that could be used to launch cyberattacks.

In DARPA's High-Assurance Cyber Military Systems (HACMS) project, research teams are investigating how to construct complex networked-vehicle software securely. An *air team* builds a software stack for unmanned aerial vehicles (UAVs), and a *ground team* investigates software for automobiles and ground-based robots. These teams are paired with a *red team* of professional penetration testers to assess the software's security vulnerabilities. The red team can access all software, design documentation, models, meeting documentation, analysis results, and system binaries produced by the other teams.

To build our air-team software securely enough to repel red-team attacks, we needed an approach that was rigorous, flexible, and compositional, to let us focus on important security concerns at several abstraction levels. As in commercial and military development, our UAVs must incorporate a significant amount of third-party software. We also expect that our UAVs could be networked to construct systems of systems whose purpose might differ considerably from the UAV system's original intent. So, we must be able to reason about requirements at various abstraction levels. Indeed, whether you consider a statement to be a requirement or design decision depends on the abstraction level on which you focus (see Figure 1).

Setting Requirements (and Their Limits)

To define meaningful requirements, we made two main assumptions about the system and potential attackers. These assumptions are essentially limits on what we can prove about the system's security.

The first assumption relates to a UAV's intended functionality and its controllability from the associated ground station. We assume that an authorized user has the authority to issue any command to the UAV, including commands that would crash or otherwise destroy it. We don't wish to limit a priori what a legitimate user may choose to do with the UAV, so we assume that all commands sent by the authorized user are legitimate. We'll only need to model whether a message

 Published by the IEEE Computer Society

(and the command it carries) is well formed. If an attacker can co-opt an authorized user's identity, no mitigation is possible.

The second assumption relates to using wireless communication. Because we can't realistically limit access to the radio spectrum, attackers will always be able to launch a denial-of-service (DoS) attack, by either jamming the physical link or overwhelming the UAV receiver with well-formed messages (even if they fail authorization). This means we can't provide absolute guarantees about reception and execution of commands from authorized users. However, we can require the UAV to reject any commands lacking authorization. We can also require the UAV to execute commands from authorized users in a timely fashion, assuming there's no DoS attack on the radio link. And, when a DoS attack is detected, our requirements can specify what actions the UAV should take to keep itself safe or avoid compromising its mission (if possible).

To construct the requirements, we followed an approach similar to that described in last issue's column, which employed Security Cards.[2] We knew quite a bit about our immediate adversary, the red team: they had strong technical skills and essentially unlimited knowledge about the system. So, we focused on a variety of known concrete attacks drawn from the Common Attack Pattern Enumeration and Classification list (http://capec.mitre.org). First, we ensured generic security principles such as user identification and authorization, secure network access and communication, secure storage, content security, and availability. From those principles, we created system-level

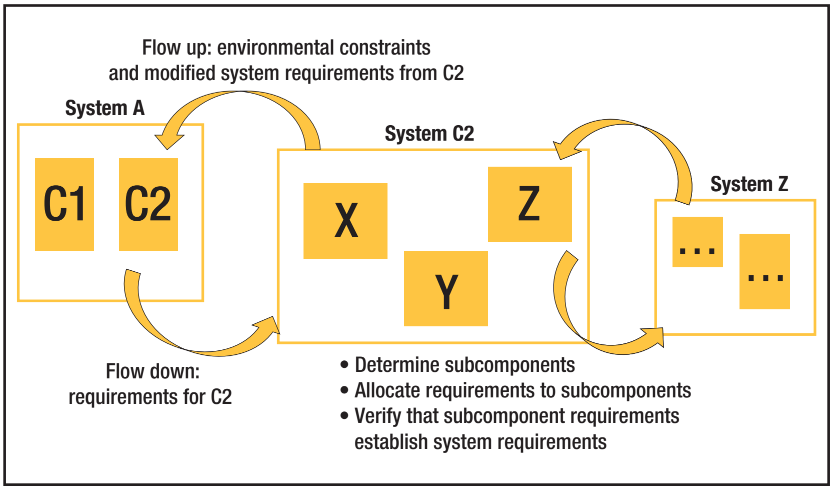


FIGURE 1. The interplay between requirements and architecture.[1] Whether you consider a statement to be a requirement or design decision depends on the abstraction level on which you focus.

security requirements for the UAV, including these:

- The UAV executes only unmodified commands from the ground station.
- If an air–ground communication link fails (or is eliminated through a DoS attack), the UAV executes its no-communication behavior.

From the system requirements, additional requirements were levied on the data link, OS, maintenance procedures, and fault handling, as well as on other system aspects.

Eliminating Weaknesses

Even with good requirements, preventing attacks is difficult; new attack methods are regularly discovered. So, we also focused on common software weaknesses that lead to security problems. The Common Weakness Enumeration website (http://cwe.mitre.org) maintains a large list of such weaknesses. Therefore, we approached the problem bottom-up, eliminating common weaknesses known to be important to many attacks, such as those related to authentication and authorization, system partitioning, maintenance, OS boot and configuration, overflow or underflow, encryption, and memory safety.

Some of these weaknesses depend considerably on the system architecture. We modeled the system architecture in AADL (Architecture Analysis & Design Language)[3] and used this model to reason about system vulnerabilities. We then constructed tools to build system images directly from the model. To ensure strict enforcement of the architectural partitioning, we used the seL4 microkernel from Data 61, which has a rigorous proof of correctness.[4]

Other weaknesses can be eliminated by the programming language. For example, Ivory, developed by Galois Inc., is an efficient domain-specific programming language that guarantees the absence of certain classes of memory errors. It also provides significant integration with

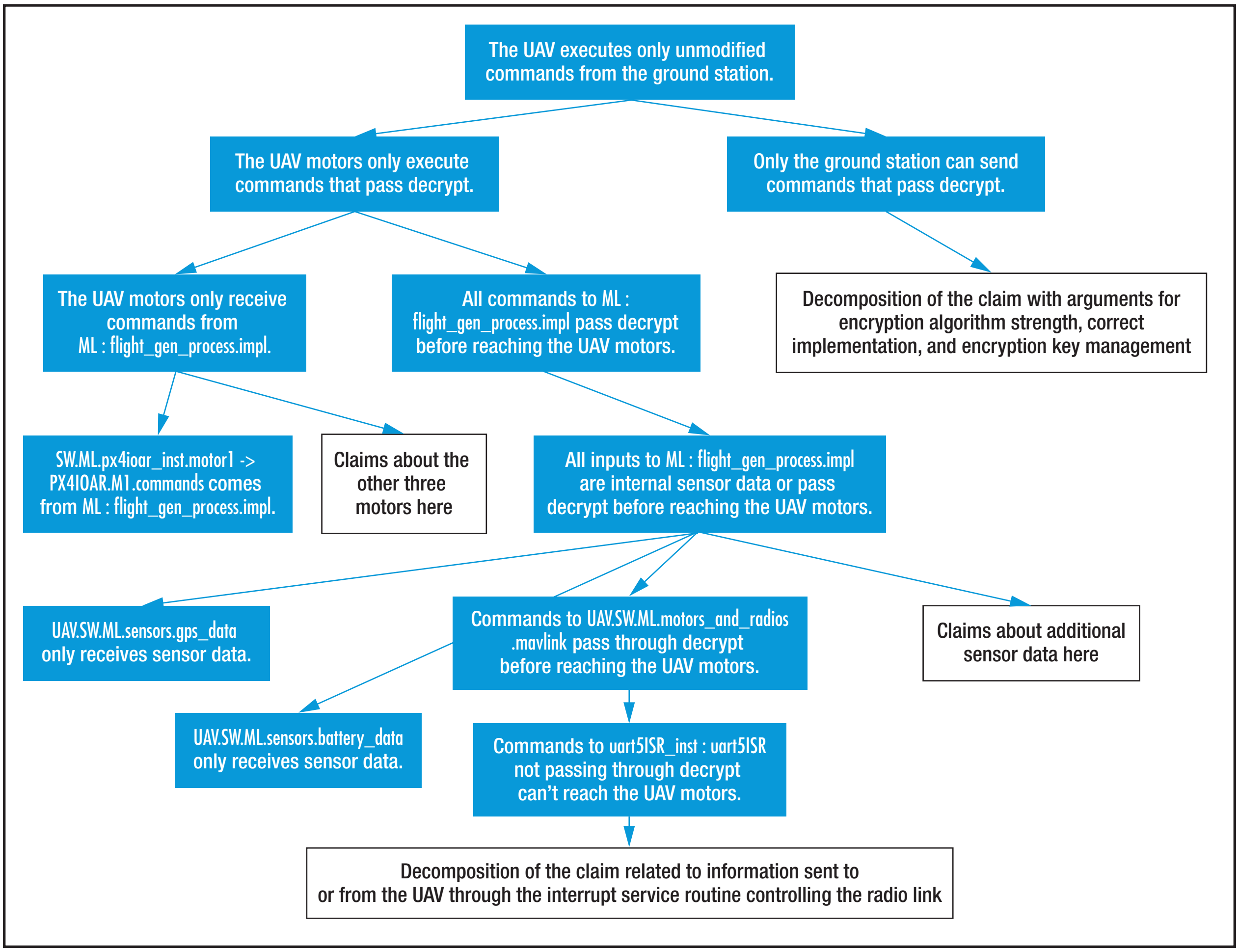


FIGURE 2. A portion of the automatically generated assurance case tree for an unmanned air vehicle (UAV) for the requirement, "The UAV executes only unmodified commands from the ground station."

model-checking tools to check for underflow or overflow, as well as cryptographic libraries.

Reasoning about Security and Composition

Assume we can define what it means for a component to be secure and can define the property Secure(A). It might be tempting to say that when we assemble components to form a larger system, we get

$$\text{Secure}(A) \wedge \text{Secure}(B) \Rightarrow \text{Secure}(A \oplus B),$$

for some (as yet undefined) composition operator $\oplus$. Such a result certainly isn't true in general, although it might hold for some specific properties and types of composition. For example, if we're concerned with a system's memory safety, it might be sufficient to demonstrate the memory safety of all that system's processes:

$$\text{MemSafe}(A) \wedge \text{MemSafe}(B) \wedge \text{MemSafe}(C) \Rightarrow \text{MemSafe}(\text{System}(A, B, C)).$$

What's more often the case in architecture-based composition is that we must look at different properties at the component level rather than the system level. In this way, composition is more like a set of lemmas used in an argument to complete a theorem's proof. Each lemma might correspond to a particular component property, a channel connecting components, or an attacker. That is,

$$\text{Lem1}(A) \wedge \text{Lem2}(\text{Chan}) \wedge \text{Lem3}(B) \wedge \text{Lem4}(\text{Attack}) \Rightarrow \text{Secure}(A \oplus B).$$

Using such reasoning, we developed an assurance case for our system's primary high-level property: "The UAV executes only unmodified commands from the ground station." Figure 2 shows a portion of the assurance case. At the top of the case, we separated this claim into two parts on the basis of the encryption setup:

- The UAV motors only execute commands that pass decrypt.
- Only the ground station can send commands that pass decrypt.

We broke down the second part into claims about our encryption protocol's strength and the keys' secrecy. The first part required a more detailed analysis of our software's architecture, including dataflow paths and memory protection.

We expressed our assurance case's general structure using logical expressions in the Resolute language.[5] Resolute lets you embed assurance case claims and rules in a system's architectural model. The Resolute engine evaluated these rules over our architecture to generate a concrete assurance case. This allowed our assurance case to adapt automatically as the model was updated and changed. Some changes could break the assurance case—for example, adding a non-memory-safe component to a nonpartitioning real-time OS; Resolute would flag such issues. We could then fix this—for example, by hosting the component in seL4 or ensuring that the component was memory-safe by compiling from Ivory. For either fix, Resolute would automatically construct the corresponding assurance case.

In addition, several portions of the architecture were assured through proof. The proof of partitioning and correct OS behavior in seL4[4] provided an ironclad foundation for building our UAV. Proofs of the Ivory type system ensured that all Ivory programs that compiled were memory-safe, which removed large classes of attacks.

The HACMS project comprises three 18-month phases. Near each phase's end, the red team receives a demonstration vehicle and software for penetration testing. In phases 1 and 2, the air team's UAVs successfully resisted all red-team attacks. In phase 1, attacks were possible only through the communications link between the ground station and UAV. In phase 2, we provided root access to a Linux partition that controlled a camera used for vehicle tracking and demonstrated that attacks launched from this partition didn't affect the UAV's flight-worthiness. In phase 3, we're adding capabilities to our UAVs, such as secure geofencing to ensure they avoid certain no-fly zones. We're also pursuing technology transfer of our tools and techniques with automotive and aerospace companies. Our experience shows that careful attention to requirements and system architecture, along with formally verified approaches that remove known security weaknesses (using Ivory and seL4), can lead to vehicles that can withstand attacks from even sophisticated attackers with access to vehicle design data.

Acknowledgments

This research was funded partly by DARPA and the Air Force Research Laboratory under the High-Assurance Cyber Military Systems project (contract FA8750-12-9-0179) and by NASA under contract NNA13AA21C.

References

1. M. Whalen et al., "Your 'What' is My 'How': Iteration and Hierarchy in System Design," *IEEE Software*, vol. 30, no. 2, 2013, pp. 54–60.
2. J. Cleland-Huang et al., "Keeping Ahead of Our Adversaries," *IEEE Software*, vol. 33, no. 3, 2016, pp. 24–28.
3. P.H. Feiler and D.P. Gluch, *Model-Based Engineering with AADL: An Introduction to the SAE Architecture Analysis & Design Language*, 1st ed., Addison-Wesley Professional, 2012.
4. G. Klein et al., "seL4: Formal Verification of an OS Kernel," *Proc. ACM Symp. SIGOPS 22nd Symp. Operating Systems Principles*, 2009, pp. 207–220.
5. A. Gacek et al., "Resolute: An Assurance Case Language for Architecture Models," *Proc. 2014 ACM SIGAda Ann. Conf. High Integrity Language Technology* (HILT 14), 2014, pp. 19–28.

MICHAEL W. WHALEN is the director of the University of Minnesota Software Engineering Center. Contact him at mike.whalen@gmail.com.

DARREN COFER is a Fellow at the Rockwell Collins Advanced Technology Center. Contact him at darren.cofer@rockwellcollins.com.

ANDREW GACEK is an industrial logician at the Rockwell Collins Advanced Technology Center. Contact him at andrew.gacek@gmail.com.

This article originally appeared in IEEE Software *vol. 33, no. 4, 2016.*

Fog Computing: Helping the Internet of Things Realize Its Potential

Amir Vahid Dastjerdi and Rajkumar Buyya, University of Melbourne

The Internet of Things (IoT) could enable innovations that enhance the quality of life, but it generates unprecedented amounts of data that are difficult for traditional systems, the cloud, and even edge computing to handle. Fog computing is designed to overcome these limitations.

The Internet of Things (IoT) promises to make many items—including consumer electronic devices, home appliances, medical devices, cameras, and all types of sensors—part of the Internet environment.[1] This opens the door to innovations that facilitate new interactions among things and humans, and enables the realization of smart cities, infrastructures, and services that enhance the quality of life.

By 2025, researchers estimate that the IoT could have an economic impact—including, for example, revenue generated and operational savings—of $11 trillion per year, which would represent about 11 percent of the world economy;[2] and that users will deploy 1 trillion IoT devices.

COPING WITH INTERNET OF THINGS DATA

IoT environments generate unprecedented amounts of data that can be useful in many ways, particularly if analyzed for insights. However, the data volume can overwhelm today's storage systems and analytics applications.

Cloud computing could help by offering on-demand and scalable storage, as well as processing services that can scale to IoT requirements. However, for health-monitoring, emergency-response, and other latency-sensitive applications, the delay caused by transferring data to the cloud and back to the application is unacceptable. In addition, it isn't efficient to send so much data to the cloud for storage and processing, as it would saturate network bandwidth and not be scalable.

Recent analysis of a healthcare-related IoT application with 30 million users showed data flows up to 25,000 tuples per second.[3] And real-time data flows in smart cities with

 Published by the IEEE Computer Society 2469-7087/17/$33.00

EDITOR SAN MURUGESAN
BRITE Professional Services;
san@computer.org

many more data sources could easily reach millions of tuples per second.

To address these issues, edge computing[4] was proposed to use computing resources near IoT sensors for local storage and preliminary data processing. This would decrease network congestion, as well as accelerate analysis and the resulting decision making. However, edge devices can't handle multiple IoT applications competing for their limited resources, which results in resource contention and increases processing latency.

Fog computing—which seamlessly integrates edge devices and cloud resources—helps overcome these limitations. It avoids resource contention at the edge by leveraging cloud resources and coordinating the use of geographically distributed edge devices.

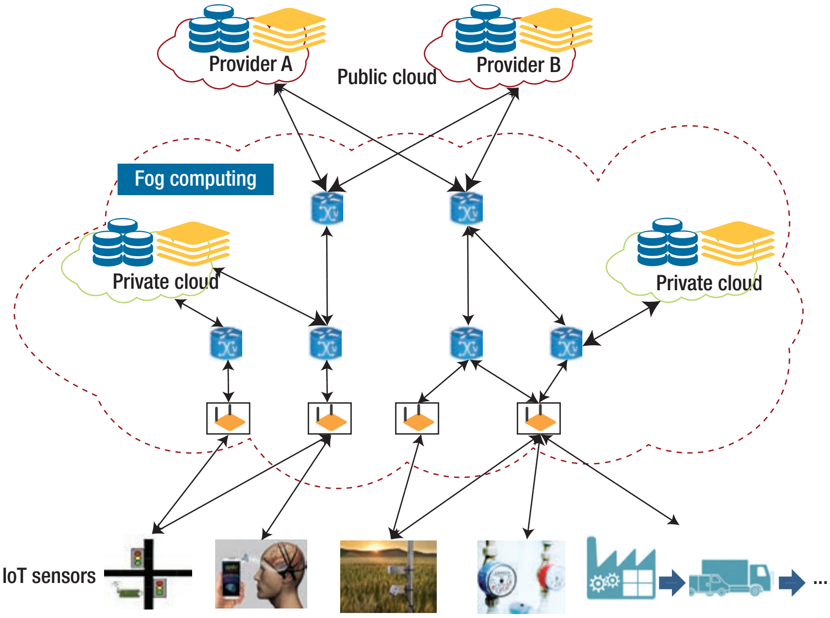


Figure 1. Distributed data processing in a fog-computing environment. Based on the desired functionality of a system, users can deploy Internet of Things (IoT) sensors in different environments including roads, medical centers, and farms. Once the system collects information from the sensors, fog devices—including nearby gateways and private clouds—dynamically conduct data analytics.

FOG COMPUTING CHARACTERISTICS

Fog computing is a distributed paradigm that provides cloud-like services to the network edge. It leverages cloud and edge resources along with its own infrastructure, as Figure 1 shows. In essence, the technology deals with IoT data locally by utilizing clients or edge devices near users to carry out a substantial amount of storage, communication, control, configuration, and management. The approach benefits from edge devices' close proximity to sensors, while leveraging the on-demand scalability of cloud resources.

Fog computing involves the components of data-processing or analytics applications running in distributed cloud and edge devices. It also facilitates the management and programming of computing, networking, and storage services between datacenters and end devices. In addition, it supports user mobility, resource and interface heterogeneity, and distributed data analytics to address the requirements of widely distributed applications that need low latency.

FOG-COMPUTING COMPONENTS

Figure 2 presents a fog-computing reference architecture. Fog systems generally use the sense-process-actuate and stream-processing programming models. Sensors stream data to IoT networks, applications running on fog devices subscribe to and process the information, and the obtained insights are translated into actions sent to actuators.

Fog systems dynamically discover and use APIs to build complex functionalities. Components at the resource-management layer use information from the resource-monitoring service to track the state of available cloud, fog, and network resources and identify the best candidates to process incoming tasks. With multitenant applications, the resource-management components prioritize the tasks of the various participating users or programs.

Edge and cloud resources communicate using machine-to-machine (M2M) standards such as MQTT (formerly MQ Telemetry Transport) and the Constrained Application Protocol (CoAP). Software-defined networking (SDN) helps with the efficient management of heterogeneous fog networks.

FOG-COMPUTING SOFTWARE SYSTEMS

There are four prominent software systems for building fog computing environments and applications.

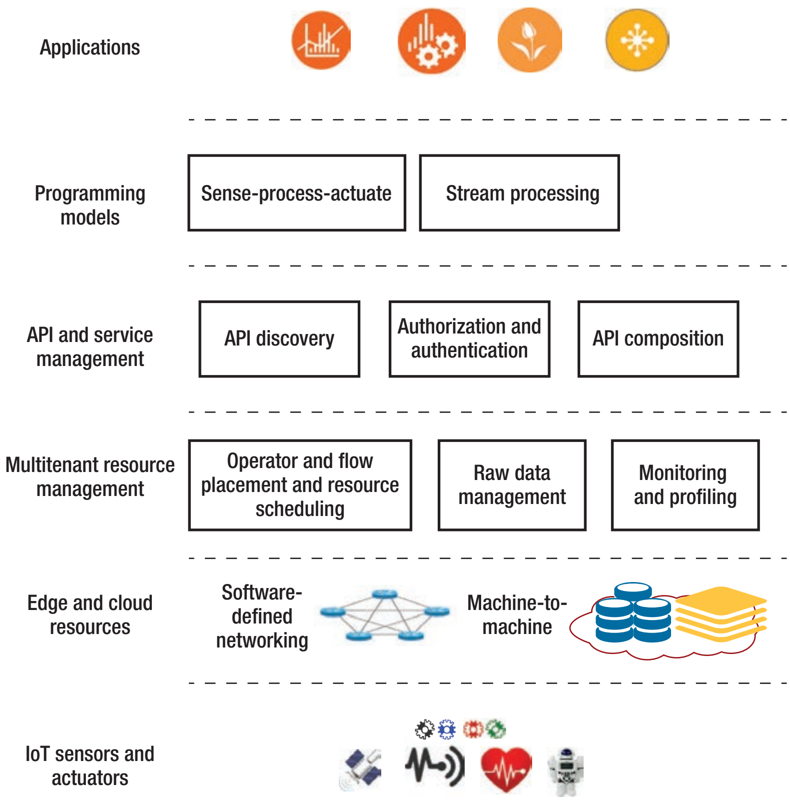


Figure 2. Fog-computing architecture. In the bottom layer are end devices—including sensors and actuators—along with applications that enhance their functionality. These elements use the next layer, the network, for communicating with edge devices, such as gateways, and then with cloud services. The resource-management layer runs the entire infrastructure and enables quality-of-service enforcement. Finally, applications leverage fog-computing programming models to deliver intelligent services to users.

Cisco IOx provides device management and enables M2M services in fog environments.[5] Using device abstractions provided by Cisco IOx APIs, applications running on fog devices can communicate with other IoT devices via M2M protocols.

Cisco Data in Motion (DMo) enables data management and analysis at the network edge and is built into products that Cisco Systems and its partners provide.

LocalGrid's fog-computing platform is software installed on network devices in smart grids. It provides reliable M2M communication between devices and data-processing services without going through the cloud.

Cisco ParStream's fog-computing platform enables real-time IoT analytics.

FOG-COMPUTING APPLICATIONS

Various applications could benefit from fog computing.

Healthcare and activity tracking

Fog computing could be useful in healthcare, in which real-time processing and event response are critical. One proposed system utilizes fog computing to detect, predict, and prevent falls by stroke patients.[6] The fall-detection learning algorithms are dynamically deployed across edge devices and cloud resources. Experiments concluded that this system had a lower response time and consumed less energy than cloud-only approaches.

A proposed fog computing–based smart-healthcare system enables low latency, mobility support, and location and privacy awareness.[7]

Smart utility services

Fog computing can be used with smart utility services,[8] whose focus is improving energy generation, delivery, and billing. In such environments, edge devices can report more fine-grained energy-consumption details (for example, hourly and daily, rather than monthly, readings) to users' mobile devices than traditional smart utility services. These edge devices can also calculate the cost of power consumption throughout the day and suggest which energy source is most economical at any given time or when home appliances should be turned on to minimize utility use.

Augmented reality, cognitive systems, and gaming

Fog computing plays a major role in augmented-reality applications, which are latency sensitive. For example, the EEG Tractor Beam augmented multiplayer, online brain–computer-interaction game performs continuous real-time brain-state classification on fog devices and then tunes classification models on cloud servers, based on electroencephalogram readings that sensors collect.[9]

A wearable cognitive-assistance system that uses Google Glass devices helps people with reduced mental acuity perform various tasks, including telling them the names of people they meet but don't remember.[10] In this application, devices communicate with the cloud for delay-tolerant jobs such as error reporting and logging. For time-sensitive tasks, the system streams video from the Glass camera to the fog devices for processing. The system demonstrates how using nearby fog devices greatly decreases end-to-end latency.

MODELING AND SIMULATION

To enable real-time analytics in fog computing, we must investigate various resource-management and scheduling techniques including the placement, migration, and consolidation of stream-processing operators, application modules, and tasks. This significantly impacts processing latency and decision-making times.

However, constructing a real IoT environment as a testbed for evaluating such techniques is costly and doesn't provide a controllable environment for conducting repeatable experiments. To overcome this limitation, we developed an open source simulator called iFogSim.[11] iFogSim enables the modeling and simulation of fog-computing environments for the evaluation of resource-management and scheduling policies across edge and cloud resources under multiple scenarios, based on their impact on latency, energy consumption, network congestion, and operational costs. It measures performance metrics and simulates edge devices, cloud datacenters, sensors, network links, data streams, and stream-processing applications.

CHALLENGES

Realizing fog computing's full potential presents several challenges including balancing load distribution between edge and cloud resources, API and service management and sharing, and SDN communications. There are several other important examples.

Enabling real-time analytics

In fog environments, resource management systems should be able to dynamically determine which analytics tasks are being pushed to which cloud- or edge-based resource to minimize latency and maximize throughput. These systems also must consider other criteria such as various countries' data privacy laws involving, for example, medical and financial information.

Programming models and architectures

Most stream- and data-processing frameworks, including Apache Storm and S4, don't provide enough scalability and flexibility for fog and IoT environments because their architecture is based on static configurations. Fog environments require the ability to add and remove resources dynamically because processing nodes are generally mobile devices that frequently join and leave networks.

Security, reliability, and fault tolerance

Enforcing security in fog environments is a key challenge due to the fact that they have multiple service providers and users, as well as distributed resources. Designing and implementing authentication and authorization techniques that can work with multiple fog nodes that have different computing capacities is difficult. Public-key infrastructures and trusted execution environments are potential solutions.[12]

Users of fog deployments also must plan for the failure of individual sensors, networks, service platforms, and applications. To help with this, they could apply standards, such as the Stream Control Transmission Protocol, that deal with packet and event reliability in wireless sensor networks.[13]

Power consumption

Fog environments consist of many nodes. Thus, the computation is distributed and can be less energy efficient than in centralized cloud systems. Using efficient communications protocols such as CoAP, effective filtering and sampling techniques, and joint computing and network resource optimization can minimize energy consumption in fog environments.

Fog computing enables the seamless integration of edge and cloud resources. It supports the decentralized and intelligent processing of unprecedented data volumes generated by IoT sensors deployed for smooth integration of physical and cyber environments.

This could generate many benefits to society by, for example, enabling smart healthcare applications. The further development of fog computing could thus help the IoT reach its vast potential. C

REFERENCES

1. R. Buyya and A. Dastjerdi, *Internet of Things: Principles and Paradigms*, Morgan Kaufmann, 2016.
2. J. Manyika et al., *Unlocking the Potential of the Internet of Things*, McKinsey & Company, 2015.
3. R. Cortés et al., "Stream Processing of Healthcare Sensor Data: Studying User Traces to Identify Challenges from a Big Data Perspective," *Procedia Computer Science*, vol. 52, 2015, pp. 1004–1009.
4. W. Shi and S. Dustdar, "The Promise of Edge Computing," *Computer*, vol. 49, no. 5, 2016, pp. 78–81.
5. F. Bonomi et al., "Fog Computing: A Platform for Internet of Things and Analytics," *Big Data and Internet of Things: A Roadmap for Smart Environments*, N. Bessis and C. Dobre, eds., Springer, 2014, pp. 169–186.
6. Y. Cao et al., "FAST: A Fog Computing Assisted Distributed Analytics System to Monitor Fall for Stroke Mitigation," *Proc. 10th IEEE Int'l Conf. Networking, Architecture and Storage* (NAS 15), 2015, pp. 2–11.
7. V. Stantchev et al., "Smart Items, Fog and Cloud Computing as Enablers of Servitization in Healthcare," *J. Sensors & Transducers*, vol. 185, no. 2, 2015, pp. 121–128.
8. J. Li et al., "EHOPES: Data-Centered Fog Platform for Smart Living," *Proc. 2015 Int'l Telecommunication Networks and Applications Conf.* (ITNAC 15), 2015, pp. 308–313.
9. J. Zao et al., "Augmented Brain Computer Interaction Based on Fog Computing and Linked Data," *Proc. 10th IEEE Int'l Conf. Intelligent Environments* (IE 14), 2014, pp. 374–377.
10. K. Ha et al., "Towards Wearable Cognitive Assistance," *Proc. 12th Int'l Conf. Mobile Systems, Applications, and Services* (MobiSys 14), 2014, pp. 68–81.
11. H. Gupta et al., *iFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in Internet of Things, Edge and Fog Computing Environments*, tech. report CLOUDS-TR-2016-2, Cloud Computing and Distributed Systems Laboratory, Univ. of Melbourne, 2016; http://cloudbus.org/tech_reports.html.
12. I. Stojmenovic and S. Wen, "The Fog Computing Paradigm: Scenarios and Security Issues," *Proc. 2014 Federated Conf. Comp. Sci. and Info. Sys.* (FedCSIS 14), 2014, pp. 1–8.
13. H. Madsen et al., "Reliability in the Utility Computing Era: Towards Reliable Fog Computing," *Proc. 20th Int'l Conf. Systems, Signals, and Image Processing* (IWSSIP 13), 2013, pp. 43–46.

AMIR VAHID DASTJERDI is a Research Fellow with the University of Melbourne's Cloud Computing and Distributed Systems (CLOUDS) Laboratory. Contact him at amir.vahid@unimelb.edu.au.

RAJKUMAR BUYYA is director of the CLOUDS Laboratory and CEO of Manjrasoft Pty Ltd., a university spin-off company that commercializes cloud-computing innovations. Contact him at rbuyya@unimelb.edu.au.

This article originally appeared in Computer *vol. 49, no. 8, 2016.*

Flying Smartphones: When Portable Computing Sprouts Wings

Ross Allen, Marco Pavone, and Mac Schwager, *Stanford University*

For decades, pop culture has imagined for us a future filled with robotic companions that attend to our daily chores. While often featured in science fiction, this vision of the future might be more accurate—and near-term—than expected. What the movies and TV shows might have gotten wrong, however, is the form of our future robotic companions. Instead of humanoids, it's aerial drones that seem to be rapidly approaching adoption for everyday tasks.

FROM PCS TO SMARTPHONES TO DRONES

At first glance, aerial drones might seem a non sequitur in the list of PCs and smartphones; however, they might indeed represent the next step-change in technology that connects the physical and digital worlds. PCs were the first technology to provide digital processing power to the average person.

Smartphones brought the next step-change in technology, because they're not only mobile but also integrate a basic set of sensors into a processing platform. Fusing a processor with a GPS receiver, accelerometers, magnetometers, and Internet connectivity has enabled so many unique applications that app developers will be exploring the design space for decades to come.

Drones mark the next leap in this progression. Along with a processor and sensor suite, drones incorporate actuators—propellers to move themselves around and (potentially) grippers to manipulate objects in the world. The fusion of these three elements (computation, sensing, and actuation), along with developments in the theory of robot autonomy,[1] allow drones to actively engage with the world around them. This is in contrast to the relatively passive interactions between humans and PCs and smartphones.

INTRODUCING SMARTDRONES

Not all unmanned aerial vehicles are consistent with a comparison to smartphones. Remote-controlled aircraft have existed for decades but, lacking any form of autonomy, can't be considered "drones" and are likely to remain strictly a hobbyist's pursuit. On the other end of the spectrum lie military drones, which are typically very expensive, complex, and highly task specific. The type of drone for which we draw parallels to smartphone technology—arguably the type that will have the most impact on the average person's daily life—are drones that we will refer to as *smartdrones*.

Smartdrones have several defining features, first of which is their *affordability*. To achieve wide-scale use, smartdrones will likely fall in the same price bracket as smartphones and modern laptops (US$500–$2,000), which makes them affordable for a common household.

The second defining feature is their *lightweight structure*. Many potential applications for smartdrones will directly or indirectly involve operations in proximity to human subjects. This immediately makes safety an important issue. Safety depends on not only a robust software architecture for autonomy but also the drone's physical design. Lighter weight, slower drones are inherently safer and thus will be the platforms of choice for operation in human-centric environments.

Another important feature is *standardization*. To enable use in a wide range of applications, smartdrones will share a quasi-standardized set of hardware and a unified control/autonomy structure. Hardware will range from components typical of smartphones

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

SMARTPHONES

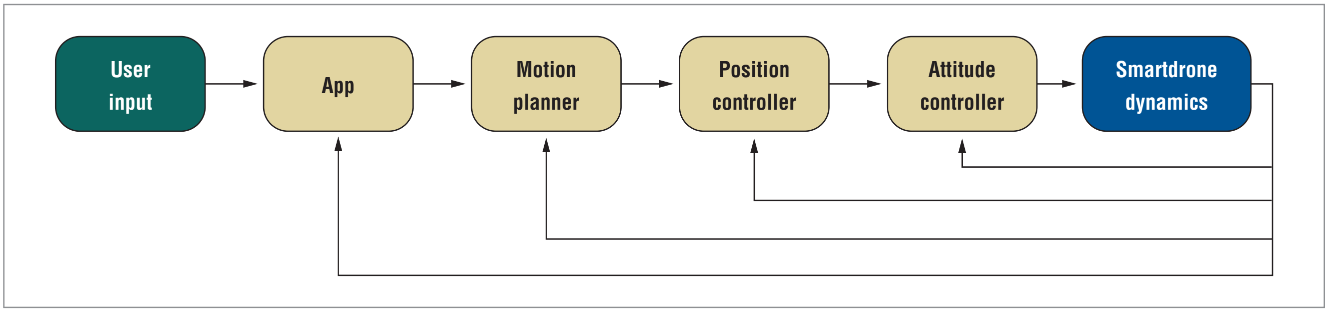


Figure 1. A high-level control/autonomy structure for smartdrone platforms.

(accelerometers, gyros, barometers, cameras, thermometers, and even microphones), to propulsive systems and manipulator/grasping mechanisms for payloads. On the software side, with a standardized GPS module and communication protocol, smartdrones will need to behave identically when it comes to avoiding restricted airspaces.

Yet arguably the most critical feature is *autonomy*. Beyond just being robust and reliable, smartdrones must also be intuitive to use. In the same way that a smartphone can be used by anyone, regardless of prior computer knowledge, smartdrones must be easily usable by those with no technical background. To achieve this, drones will have to be highly autonomous—relying on the onboard processor for all low-level control and leaving only application selection and a few input options to the user.

Note that thus far, we have made no distinction between quadrotors and fixed-wing aircraft when referring to smartdrones. This is because both can meet our definition, so either can be adopted. The hover capabilities of quadrotors tend to add an additional safety layer over fixed-wing aircraft, making them the more attractive option for applications in human proximity. Fixed-wing aircraft, on the other hand, offer much greater range and endurance. Hybrid craft, such as tiltrotor craft, would also fit our definition of smartdrones.

DEVELOPING AUTONOMOUS DRONES

Hereafter, we further focus on the autonomy feature, reviewing the technology itself, its safety aspects, and the range of applications it enables.

Understanding the Control/Autonomy Structure

A unified control/autonomy structure is key to the smartdrone concept, so app developers will know that the software they develop will interact with firmware and hardware in much the same way, regardless of the smartdrone model or manufacturer (similar to how an app can be released on Android or iOS with little additional work). The unified control/autonomy structure will likely mirror the structure that has been developed for many research-based quadrotors (see Figure 1).

The control structure is composed of a set of nested loops. Outer loops, responsible for more abstract decisions, feed information down to inner loops, usually in the form of *setpoints* or *reference targets*, which drive the direct control of the smartdrone hardware. Sensors provide feedback about the state of the smartdrone to the relevant control layer.

Specifically, the user selects an application for the smartdrone, and the app produces a set of high-level objectives. The motion planner fuses these objectives with information about the world—such as obstacle locations, no-fly zones, or speed restrictions—to come up with a feasible plan for achieving the objectives. The position controller is tasked with executing the plan by comparing the desired position from the motion planner with the actual position read by the sensors and performing feedback control.

The attitude controller is tasked with stabilizing the aircraft and executing the positioning commands from the position controller. Because most drone platforms are underactuated, the attitude controller is a "slave" to the position controller in that arbitrary positions and velocity can't be achieved independent of attitude, so the attitude controller accommodates the desired positions and velocity. While the outer loops of the control structure will employ sophisticated optimization, control, and decision-making techniques, the inner loops will likely apply simple yet robust proportional-integral-derivative controllers.

Ensuring Safety

As PCs, smartphones, and smartdrones introduce progressively more powerful technological applications, they also carry an ever increasing burden of risk—an example of the proverbial double-edged sword. For example, PCs allowed average people to digitize most of their personal credentials and financial information. This greatly simplified tedious tasks such as filing taxes, but it also opened the door to risks such as identity fraud. The primary safety aspects that are being addressed as smartdrones are adopted in wide-scale use fall in the categories of sensing, planning, verification, and system-level integration.

Sensing. Each layer of the smartdrone control/autonomy structure requires its own sensing hardware, which is used to estimate the current state of the craft. The innermost layer—the

Figure 2. Demonstrating real-time kinodynamic planning on a quadrotor dodging a fencing blade.

attitude controller in Figure 1—is, for the most part, a solved problem. Even inexpensive, off-the-shelf inertial measurement units are sufficient to estimate and control attitude. This is why you can purchase a quadrotor "toy" for less than $50 and have it hover and perform basic motions. Such remote-controlled toy quadcopters are often well trimmed, so they can hover in place fairly reliably. Being well-balanced to avoid drifting during hover is, however, very different from autonomously controlling the drone's position.

Position control—the second innermost layer in Figure 1—presents a greater challenge because position estimation requires considerably more sophisticated hardware than that of attitude control. For absolute position, a smartdrone would require a GPS module. Such modules are relatively expensive, running at $80 for a hobby-grade component. Furthermore, GPS alone might be insufficient to guarantee safe operation. GPS relies on line-of-sight to GPS satellites, making it unreliable in environments with partial or full obstruction of the sky (as can occur in canyons, in forests, near tall buildings, or indoors). Thus, smartdrones will likely have to supplement GPS information with localized position information to provide terrain-relative position data. Several sensor types are capable of achieving this, including sonar, light detection and ranging (lidar), and vision sensors. In the end, position estimation will be achieved using a mixture of these technologies.

Planning. Even with perfect and complete sensor data and an infallible controller, a major issue exists in how to decide what trajectory a smartdrone should take through a complex, dynamic world. These questions have been at the center of the field of robot motion planning for years.

Smartdrones present a particularly challenging form of robot motion planning, because they require the consideration of a high-speed robot in a changing environment. This form of motion planning, termed real-time kinodynamic planning, is an active field of research. Recent work at the Autonomous Systems Laboratory at Stanford University has developed a framework for solving such problems in real time.[2] The framework operates on an offline-online computation paradigm, whereby a library of trajectories is precomputed offline and then efficiently pruned online when environment data becomes available. Machine learning and optimal control techniques make such a procedure fast and accurate, in the sense that near-optimal trajectories are repeatedly computed every few milliseconds. Figure 2 shows the application of such a framework to the control of a quadrotor that dodges a fencing blade.

Verification. Recent work has sought to verify the safety of smartdrone systems by embedding verification directly into the design of the control/autonomy module. The field of formal methods, which was traditionally developed to verify the correctness of computer programs, has now been applied to design drone control systems that are correct "by construction." For example, work at the Stanford Multi-Robot Systems Lab and the Boston University Robotics

SMARTPHONES

Lab has led to formal methods algorithms that construct provably safe trajectories for multiple smartdrones to perpetually monitor an environment, while scheduling sufficient time to recharge their own batteries.[3]

System-level integration. The US Federal Aviation Administration (FAA) has recently overhauled its regulations regarding the use of Unmanned Aircraft Systems. It now requires private operators to register their drones in a national database, and it prohibits the flying of recreational drones near airports and other areas with sensitive airspace. As drone capabilities grow, and as autonomous features find their way into commercialized drone technology, the FAA is incrementally taking steps to integrate drones into the already complex US airspace (www.faa.gov/uas). Likewise, the European Aviation Safety Agency is taking precautions to integrate private drone usage safely into the European airspace (www.easa.europa.eu/easa-and-you/civil-drones-rpas). Nonetheless, significant challenges remain to safely integrate drones—whether commercial or private—into the airspace, and the regulations governing drone usage are expected to evolve considerably in the coming years.

Reviewing Smartdrone Apps

Some of the first applications for quadrotors (and more generally, smartdrones) were realized in the laboratory. Due to their ease of control and their robustness to changing configurations, quadrotors became an excellent demonstration platform for navigation, planning, and network algorithms.[4] These research demonstrations paved the way for many of the commercial and military applications we see being developed today.

Perhaps the application that has received the most attention by the public is the proposed use of unmanned aerial delivery platforms such as Amazon Prime Air. Drone delivery has the potential to radically change the way we access consumer products, because it would reduce the delivery time for online purchases from days to minutes. Amazon's delivery system doesn't quite match our description of smartdrones, however, because it involves expensive, large, task-specific aerial robots; not flexible, inexpensive platforms usable by the public. Restaurants, on the other hand, could use the more universal smartdrone concept for delivery of small food items to local communities. Similarly, medicine and first-aid supplies could be delivered to remote or hazardous areas during disaster events.

Another drone application that has made its way into mainstream media is that of recreation use, specifically for action/adventure sports. Established drone companies such as DJI, and startups such as Lily Robotics, are planning to offer multirotor aircraft that are designed to autonomously follow a user and shoot video. Although these products aren't yet on the market, the significant number of companies and startups pursuing this concept gives credence to the idea that we will soon see quadcopters chasing skiers down mountains.

Currently, the most economically viable—albeit lesser known—application for drone technology lies in agriculture. Companies such as 3DR are providing autonomous multirotor craft that can survey crops by recording multispectral images of farmland.

The power of the smartdrone concept becomes even more apparent when you imagine collective multidrones acting collaboratively to carry out large-scale tasks. Just as the benefits of smartphones have exploded with the advent of mobile apps for social networking that cull data from a collection of users, the capabilities of smartdrones will explode as the interconnectedness of the drone network increases. Today, mobile apps that mine data from hundreds of thousands of daily users, such as Waze and Tealeaf, can effectively predict phenomena as diverse as traffic and stock prices. Tomorrow, smartdrones will leverage the perpetual networked aerial drone presence to give rich, real-time data about agricultural crops, traffic, weather, the movement of wildlife, and the activities of suspected criminals, giving early warnings for everything from wildfires to freeway pileups.

Furthermore, many of the deficits of the small size of smartdrones, including limited flight time, range, and payload, can be alleviated when you consider the coordinated actions of large groups of drones. A thriving research community in multirobot systems and multiagent control is currently devoted to solving problems of large-scale coordinated autonomy. New decentralized algorithms are emerging for control, perception, and trajectory planning over a wireless network to enable multidrones systems: groups of drones that reach collective inferences about the world and make collective decisions about what actions to take in the world to accomplish a task.

The potential applications of collective smartdrones are vast. Perhaps the first capability that will be realized is large-scale distributed perception. Drones will provide us with a perpetual sensor network in the sky to sense diverse forms of data for a variety of purposes,[5] as Figure 3 illustrates. As mentioned, farmers are already using individual drones for crop sensing to see daily or weekly detailed snapshots of crop health. These snapshots then inform decisions about watering, fertilizing, and applying pesticide to specific areas of the crops where most needed. With the advent of smartdrone networks, farmers could have an on-demand updated computer model of the health of their crops for crop management decisions.

Smartdrone networks will also help search-and-rescue teams find lost hikers in the wilderness, or victims of boating accidents lost at sea. The key is the ability of a smartdrone network to parallelize the task of gathering information over a large area. The larger the area, the more

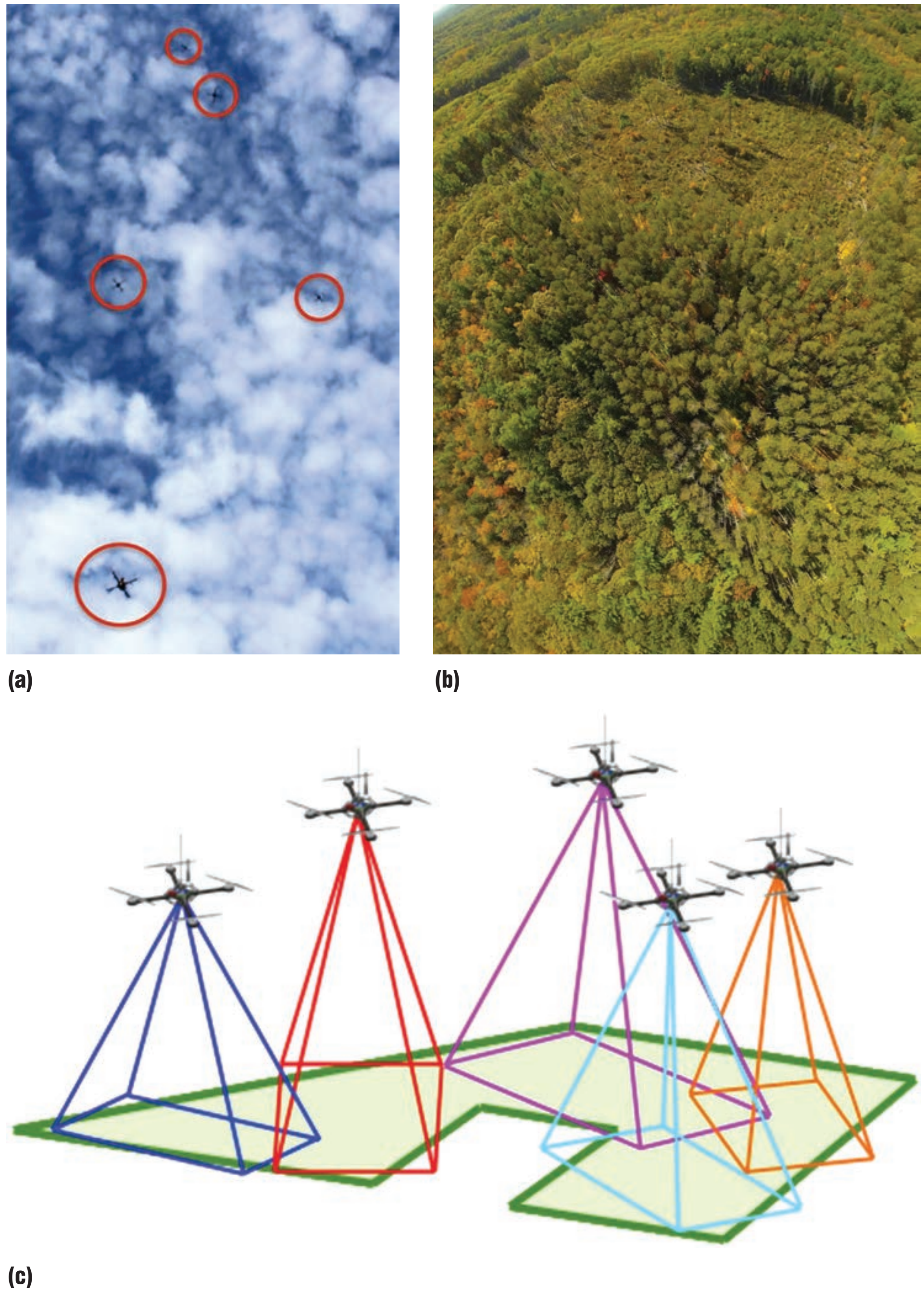


Figure 3. Five smartdrones cooperatively surveying a research forest: (a) a perpetual sensor network in the sky can (b) survey the forest below, (c) parallelizing the task of gathering information over a large area.

drones you can deploy to search it efficiently. Construction sites, which are frequently targeted for theft, and large-scale infrastructure, which requires frequent inspection, could employ smartdrones for persistent surveillance. High-tech border security could implement a fleet of smartdrones that could monitor large stretches of remote terrain.

Beyond merely sensing the environment, smartdrones interacting with the environment (for example with grippers, display lights, and other actuators) will open up a new range of exciting applications. For example, a group of smartdrones with colored LEDs can form a massive 3D display,

SMARTPHONES

creating a new medium for art, entertainment, communication, and marketing. Researchers at ETH Zurich have already shown the promise of such drone displays.[6]

When equipped with grippers, a smartdrone collective might soon replace cranes in construction sites, collaborating to hoist heavy beams into place to build buildings and bridges alongside human construction workers.[7] One day, national forests might employ groups of autonomous smartdrones to not only monitor for forest fires but also fight such fires with the targeted application of fire retardant. In addition, farmers might use smartdrones to not only monitor crop health but also actively manage crops by applying water, fertilizer, and pesticide with surgical precision. Indeed, the most transformative applications for smartdrones are most likely still waiting to be discovered by the app developers and drone users of the future.

Due to their ability to actively and autonomously interact with the world, lightweight, highly autonomous drones are emerging as the next step-change in consumer electronic technology, much in the same way that smartphones revolutionized personal computing. Although research is ongoing to ensure safe, autonomous operation, smartdrone systems are already being used in several applications, with many more applications soon to emerge. After two decades of research and development, portable computing has finally sprouted wings!

REFERENCES

1. *Handbook of Unmanned Aerial Vehicles*, K.P. Valavanis and G.J. Vachtsevanos, eds., Springer, 2015.
2. R. Allen and M. Pavone, "A Real-Time Framework for Kinodynamic Planning with Application to Quadrotor Obstacle Avoidance," *Proc. AIAA Conf. Guidance, Navigation and Control*, 2016; http://arc.aiaa.org/doi/10.2514/6.2016-1374.
3. K. Leahy et al., "Persistent Surveillance for Unmanned Aerial Vehicles Subject to Charging and Temporal Logic Constraints," *Autonomous Robots*, Nov. 2015, pp. 1–16.
4. D. Mellinger, N. Michael, and V. Kumar, "Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrotors," *Int'l J. Robotics Research*, vol. 31, no. 5, 2012, pp. 664–674.
5. M. Schwager et al., "Eyes in the Sky: Decentralized Control for the Deployment of Robotic Camera Networks," *Proc. IEEE*, vol. 99, no. 9, 2011, pp. 1541–1561.
6. J. Alonso-Mora et al., "Object and Animation Display with Multiple Aerial Vehicles," *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems*, 2012, pp. 1078–1083.
7. Q. Lindsey, D. Mellinger, and V. Kumar, "Construction with Quadrotor Teams," *Autonomous Robots*, vol. 33, no. 3, 2012, pp. 323–336.

Ross Allen is a PhD candidate in the Department of Aeronautics and Astronautics at Stanford University. Contact him at rallen10@stanford.edu.

Marco Pavone is an assistant professor in the Department of Aeronautics and Astronautics at Stanford University. Contact him at pavone@stanford.edu.

Mac Schwager is an assistant professor in the Department of Aeronautics and Astronautics at Stanford University. Contact him at schwager@stanford.edu.

This article originally appeared in IEEE Pervasive Computing *vol. 15, no. 3, 2016.*

Susanne Boll
University of Oldenburg, Germany

Eye-Controlled Interfaces for Multimedia Interaction

Chandan Kumar, Raphael Menges, and Steffen Staab
Institute for Web Science and Technologies (WeST)

In the digitized world, interacting with multimedia information occupies a large portion of everyday activities; it's now an essential part of how we gather knowledge and communicate with others. It involves several operations, including selecting, navigating through, and modifying multimedia, such as text, images, animations, and videos. These operations are usually performed by devices such as a mouse or keyboard, but people with motor disabilities often can't use such devices. This limits their ability to interact with multimedia content and thus excludes them from the digital information spaces that help us stay connected with families, friends, and colleagues.

When we interact with multimedia—which presents rich visual content—our eyes process relevant information. Eye activities can be tracked, and gaze tracking has been successfully used to analyze and evaluate cognition during multimedia learning.[1] So far, however, tracking of this visual channel has rarely been exploited to control interaction with multimedia information. Seminal work has started to employ eye-tracking technology for controlling interactions, targeting two major paradigms for interaction by eye tracking:[2] direct control and implicit observations.

Direct control refers to the deliberate, explicit use of eye movements by the user with the intention that such gaze signals are picked up for interaction commands, such as selecting, moving, or modifying an object or defining a new one—by text input, for example. *Implicit observations* of gaze signals have been used to enhance the viewing activity (such as for extended reading, as in Text 2.0[3] or to capture the varying importance of multimedia visual content[4]).

Direct control is effective but very slow and tiring for the human user. Implicit observations remain unobtrusive but can at best be considered a weak signal of what the user wants to accomplish. Thus, direct control through gaze tracking should be supplemented and merged with interactions derived from implicit observation to remain effective while improving the user friendliness of eye-tracking technologies in multimedia interaction. In this context, the EU-funded MAMEM project (Multimedia Authoring and Management using your Eyes and Mind) aims to propose a framework for natural interaction with multimedia information for users who lack fine motor skills (see the related sidebar for more information).

Here, we primarily focus on the gaze-based control paradigm (see Figure 1) that we've developed as part of our work at the Institute for Web Science and Technologies (WeST) within the scope of MAMEM project. We outline the particular challenges of eye-controlled interaction with multimedia information, including initial project results. The objective is to investigate how eye-based interaction techniques can be made precise and fast enough to not only allow disabled people to interact with multimedia information but also make usage sufficiently simple and enticing such that healthy users might also want to include eye-based interaction.

Challenges of Eye Input

Eye-tracking systems measure a person's eye movements so that the gaze point is established at any point in time. Different invasive or noninvasive methods for eye movement measurement have been investigated to improve gaze data estimation.[2] Eye-tracking technology has evolved, increasing precision and decreasing cost. However, using an eye gaze as input remains a challenge, due to the limitation of the visual angle, calibration errors, the drift, and inherent eye jitter, as well as the fact that the gaze positions reported by eye trackers don't correspond exactly to where the user is looking.

Published by the IEEE Computer Society

Multimedia Authoring and Management Using Your Eyes and Mind

The MAMEM project started in May 2015, and it consists of eight different partners (see www.mamem.eu/project/consortium) collaborating to deliver the technology that will let people operate software applications and execute multimedia-related tasks using their eyes and mind. MAMEM especially targets individuals with motor disabilities (such as people with Parkinson's disease, muscular disorders, and tetraplegia). The common symptom of these disorders is the loss of the voluntary muscular control (while preserving cognitive functions), leading to a variety of functional deficits, including the ability to operate applications that require the use of a conventional interaction medium (mouse, keyboard, touchscreens, and so on). As a result, the affected individuals are marginalized and can't keep up with the rest of the society in a digitized world.

MAMEM's aims to better integrate these people into society by endowing them with the critical skill of accessing multimedia information content using novel and more natural interface channels. MAMEM also aims to make its technology persuasive and provide the principles for designing interfaces that will motivate disabled people to use them.

Eye movements occur as sequences of very fast (less than 100 ms) saccades, followed by relatively stable fixation periods (100–600 ms). Even during fixations, however, the eye isn't completely still but is characterized by some jitters. In addition, even the precision of recent eye trackers is still limited, because the trackers can only determine the angle of the user's gaze in relation to the screen up to approximately 0.5 degrees (or roughly 10 pixels on a 17-inch display observed from 60 cm away). These factors imply a loss of stability, causing the "eye cursor" to momentarily leave the target intended by the user for further multimedia interaction. *Direct control* is impeded by such factors. Consequently, for reliable eye-controlled interfaces, smoothing mechanisms are necessary to stabilize the gaze signal. Furthermore, interface elements must be adopted to negotiate the impact of limited accuracy.

Another major difficulty for input control by gaze interaction is the double duty performed by eyes. When using an eye tracker for control, the "normal" course of events changes substantially, because the eyes are both acting as an important sensory channel and providing motor responses to control the computer.[5] Instead of the hand providing motor responses to control the computer through external physical devices, the eye provides motor response through virtual or graphical controls that appear on the system's display. The eyes are thus overloaded with explicit control tasks, especially in complex scenarios of multimedia exploration and modification, making the interaction tedious and error-prone. Therefore, it's imperative for eye-controlled interfaces to use implicit eye signals to predict user intentions and support the explicit control of tasks.

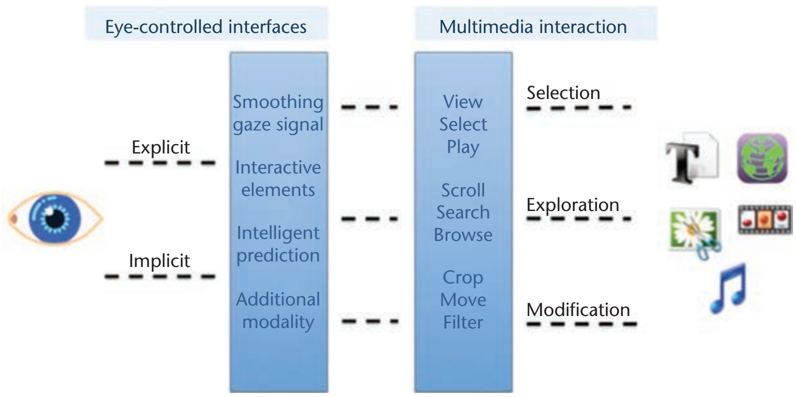


Figure 1. Eye-controlled interfaces for multimedia interaction. Examples include selecting, exploring, and modifying text, images, and videos. (Note that interaction with multimedia isn't limited to this set of operations.)

Furthermore, eyes engage in inspection and selection simultaneously while interacting with the interface. The most common method to distinguish between inspections and selections is to set a time threshold (or *dwell time*), with a click issued after the duration of the fixation exceeds a specified amount of time. The dwell time might lead to unintended activations, resulting from fixations used for inspection being confused with a selection. This issue is referred to as the *Midas-Touch problem*.[2] Appropriate visual feedback from the eye-controlled interfaces could play a vital role in helping users discriminate between inspection and selection. Moreover, an additional source of input confirmation could also be a significant measure for distinguishing between selections and inspections.

To address these challenges and increase the feasibility of eye-based interactions, we employ

> **Another significant element of eye-controlled interfaces is providing feedback to users with respect to their gaze activity.**

the following four techniques: we smooth gaze signals to counteract eye jitter, design novel interface elements for eye-controlled interaction, add implicit intelligence to gaze-centered interfaces, and integrate additional input modalities.

Smoothing Signals

Researchers have pointed out the necessity of diminishing the effects of eye jitter to improve the stability of the eye cursor using smoothing. Oleg Špakov compared several eye-movement filters for use in HCI applications.[6] The comparison was based on the introduced delay, smoothness, and closeness to the idealized data. The study's outcome was that algorithms with state detection (fixation and saccade) and adapted processing generally performed better than others. To smooth the data from the eye tracker in real time, it's necessary to determine whether the most recent data point is the beginning of a saccade, a continuation of the current fixation, or an outlier relative to the current fixation. The x and y components of the raw gaze points are mapped according to two independent, linear functions.

For smoothing, we can also apply moving averages—that is, the average location of every k successive gaze points within a fixation window, where the fixations and saccades are treated separately. Manu Kumar and his colleagues presented a one-sided triangular filter to compute the fixation point as a weighted mean in the current fixation window.[7] They also applied two Kalman filters to process the eye-gaze data, one for the entire raw gaze data, and the other for the gaze data within fixation windows.

In addition to the smoothing effect, we can generate information about the start and end of fixations. This can be used to generate high-level gaze events. The algorithm proposed by Darius Miniotas and his colleagues, the *grab-and-hold algorithm,*[5] had the same effect as if the gaze were held on the desired target during periods of fixation, thus effectively reducing the probability of restarting the selection timer before the end of the dwell time.

Designing Interactive Interface Elements

Interfaces generally use their architecture to acclimatize to the type of input device used for operating the computer. The look and feel of the interface depends on the device selected for primary input. For example, when a mouse or keyboard is the primary input device, the interface-controllable elements, such as buttons, icons, menus, scrollbars, lists, and dialog boxes, will appear as they do in a conventional interface. However, when a different physical input device, such as an eye tracker, is the primary source, the look and feel of these elements must change to accommodate eye input.

Facilitating Accuracy

As noted, the gaze position acquired from eye-tracking devices doesn't exactly correspond to where the user is looking. This problem can be addressed with interface adaptations—such as enlarging targets when the eye-gaze interaction involves acquiring small targets. Moreover, when the user looks at the screen, the area around the gaze point can be linearly magnified and redisplayed in a zoom window.[8] A magnified view helps map gaze points to a desired target, so the desired actions can be easily and correctly performed. Distorting the gaze area around the desired target can also help systems better understand the user's gaze.

Another way to magnify a target is to temporarily expand the target itself rather than to zoom in on the area around it. When the user's gaze falls within the vicinity of the desired target, the target size could increase to include the gaze point into the enlarged target.

Visual Feedback

Another significant element of eye-controlled interfaces is providing feedback to users with respect to their gaze activity.[9] Most errors are induced by the lack of adequate feedback from the screen, because the slightest discrepancy between a user's eye movements and what he or she sees, feels, or hears can disrupt the experience. Designers thus must repurpose the feedback mechanisms for sensory information from eyes.

Using adequate visual graphics and animation as feedback could help users discriminate between inspections and activations and could reduce errors in interactive operations—for example, buttons get activated when the gaze hits them, and they shrink after activation to trigger the button. A colored overlay increasing in size would work as a visual representation of the remaining time until the trigger. Furthermore, the visual feedback shouldn't be unanimous for all kind of operations—that is, a scroll button should have a specific visual feedback indicating the page lengths that have been scrolled.

Figure 2 shows some examples of these elements. The top row in Figure 2a signifies a click emulation through eye-gaze interaction, where the animated highlighting over the icon shows the gaze duration, and the click is activated like a switch button. The bottom row indicates a sensor-like button, relevant for progressive elements like scrolling. Figure 2b shows the stages of eye typing, with the magnifying effect of character selection combined with the visual feedback of the user's gaze.

The discussed accuracy and interactive elements become essential to develop applications that use the eye as a direct control for interaction with multimedia information. In that regard, we have employed these heuristics in various multimedia interaction environments.

Figure 3 shows an example of gaze-adapted browsing (with GazeTheWeb), where the conventional browser interface is customized for eye gestures. The custom layout on the left and right indicates the enlarged graphical elements to select various browsing operations and visual feedback for interacting with these elements. The major central region is the Web view containing the content of the webpages. The left layout contains browser-centric functionality, such as opening new tabs, going backward and forward, and changing various browser settings. The right layout is for interacting with the webpage. The user can view, scroll, and navigate through the image and text content though gaze direction. To rapidly scroll up and down, dedicated buttons are used that act like sensors, where the visual saturation increases when the user looks at the buttons.

Clicking on images and text hyperlinks and navigating to different pages is an essential

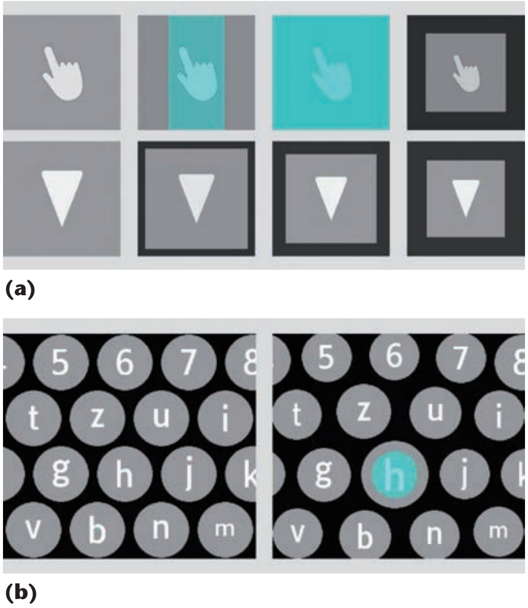


Figure 2. States of activation for interactive elements of eye-controlled interfaces: (a) visual feedback and (b) visual feedback with magnification on the right side of the screen.

component of a user's Web-browsing behavior. For such navigational task, links might be very close to each other. Consequently, we applied the strategy of dynamically magnifying the observed screen portion when the user is looking at an intended hyperlink, so the link can be accurately selected within the enlarged page region. Similarly, Figure 4 shows the gaze-adapted Twitter application interface, where the user can perform all essential operations (such as tweeting, searching, following, or discovering) interactively via direct control through the eyes.

Figure 5 shows the example of an eye-controlled interface for the game "Schau genau!" where gaze is used to control a butterfly. The player collects flowers and classifies photographs of flowers to earn points (see Figure 5a).[10] In this immersive game environment, several interaction elements were included with respect to the button size, shape, visual feedback, and so on. For example, Figure 5b shows the game screen space for the player inserting his name for his high score. The user can scroll horizontally through letters of the alphabet. The fixated letter enlarges until a dwell time is over and the letter is selected. If the player fixates another letter in the meantime, the old one is scaled back down. On the bottom of the game screen, the

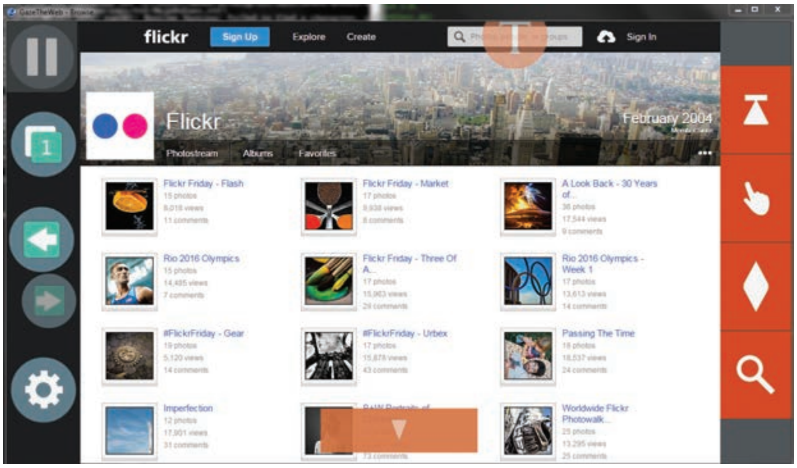


Figure 3. Eye-controlled Web browsing. Here, the user is exploring a collection of images.

Figure 4. Eye-controlled social media browsing. Here, the user can select and interact with tweets.

player can either confirm the input or delete the latest written letter using individual selection buttons.

The interfaces in Figures 3–5 were well received by users (see the video demonstrations of the applications at https://west.uni-koblenz.de/en/research/mamem). The Schau genau! game was installed in a horticultural garden show in Landau (http://lgs-landau.de) in a stand-alone arcade cabinet. The game was played more than 2,900 times during the exhibition—a clear indication of its popularity and the acceptability of its adopted interaction elements. The Twitter interface shown in Figure 4 was highly appreciated by users in a lab study with 13 participants. The eye-controlled interface was compared against the conventional method of emulating a "mouse with eyes" (OptiKey; https://github.com/OptiKey/OptiKey/wiki), and it significantly outperformed the eye-mouse emulation in the metrics of system usability and mental workload.

Inspired by the success of these design and interaction elements, we proposed an *eyeGUI* library to design and develop GUIs suitable for eye-based input control. The eyeGUI library enables the manipulation and rendering of user interfaces for eye-tracking input.[11] A variety of elements, such as buttons, images, and text, can be used from the library to build a proper interface. All elements in eyeGUI are designed especially for eye tracking in terms of their size, appearance, and user interaction—for example, buttons get activated when the gaze hits them, and they shrink after activation to trigger the button. The eyeGUI library was developed in C++ 11 and is based on OpenGL. You can use it to build user interfaces for eye tracking by adding XML files as layouts and manipulating elements within these layouts via "listeners." The listeners can be accessed in the application environment to give every interface element its own functionality.

Adding Implicit Intelligence

Predicting a user's intent via implicit gaze signals can help enhance control functionality. Here, we describe how implicit observations might help with three example activities—scrolling, image search, and editing.

Scrolling

The act of scrolling is strongly coupled with the user's ability to engage with information via the visual channel. Therefore, the use of implicit eye-gaze information is a natural choice for enhancing multimedia content scrolling techniques.

For scrolling and reading, we propose using natural eye movements to control the motion of the windows for the user. In the GazeTheWeb interface, the user must explicitly activate automatic scrolling, turning it on and off by pressing a button using direct gaze control (see the right side of Figure 2a). In the auto scroll mode, the user has a smoother and more natural reading experience, because the scrolling is supported via implicit observation of user's gaze coordinates. The scroll direction is determined by noting the quadrant where the user is currently looking, outside a central neutral region. The scroll speed is proportional to the distance from the center of the screen.

Searching Images

Another focus of our work is to employ implicit gaze pattern for an improved search experience—that is, to make the GazeTheWeb interface more fluent and natural by collecting the feedback implicitly while users are involved in gaze-based browsing. Essentially, the user implicitly requests better images by zooming in, and this feedback is inferred from gaze data while the user looks at the images. This could significantly reduce the user's effort for explicitly refining search results.[4]

There have been some preliminary studies related to the use of implicit gaze information in image retrieval.[12] However, GazeTheWeb provides a more natural scenario of gathering implicit feedback to enhance the results while users are actively engaged in gaze-driven searching and browsing. For example, Figure 6a shows an image search scenario in which the user is browsing through results after searching for "plants." The user zooms in on the pictures for an enlarged view, providing implicit feedback on the region of interest. Based on the fixations and user attention, the system goal is to present refined results to support the user's interest (the desert plants shown in Figure 6b).

Editing

Gaze-based browsing not only offers a framework for eye-controlled access but also recognizes relevant signals regarding the user's region of interest. We plan to use such fixation data to enhance multimedia interaction—for example, to identify important content and use it for multimedia editing (such as image cropping). With the GazeTheWeb interface, users simply look at images while browsing, and we can use the gaze patterns to identify the important image content and automatically generate crops of any size or aspect ratio. The goal is to create appealing crops without explicit interaction. Furthermore, precise identification of relevant image content without explicit interaction is a vital feature. It lets us analyze and quantify the viewing behavior of images and how users select the region of interest for image editing. It also lets us analyze other useful functionalities, such as the automatic creation of snapshots or of thumbnails for adaptive Web documents.

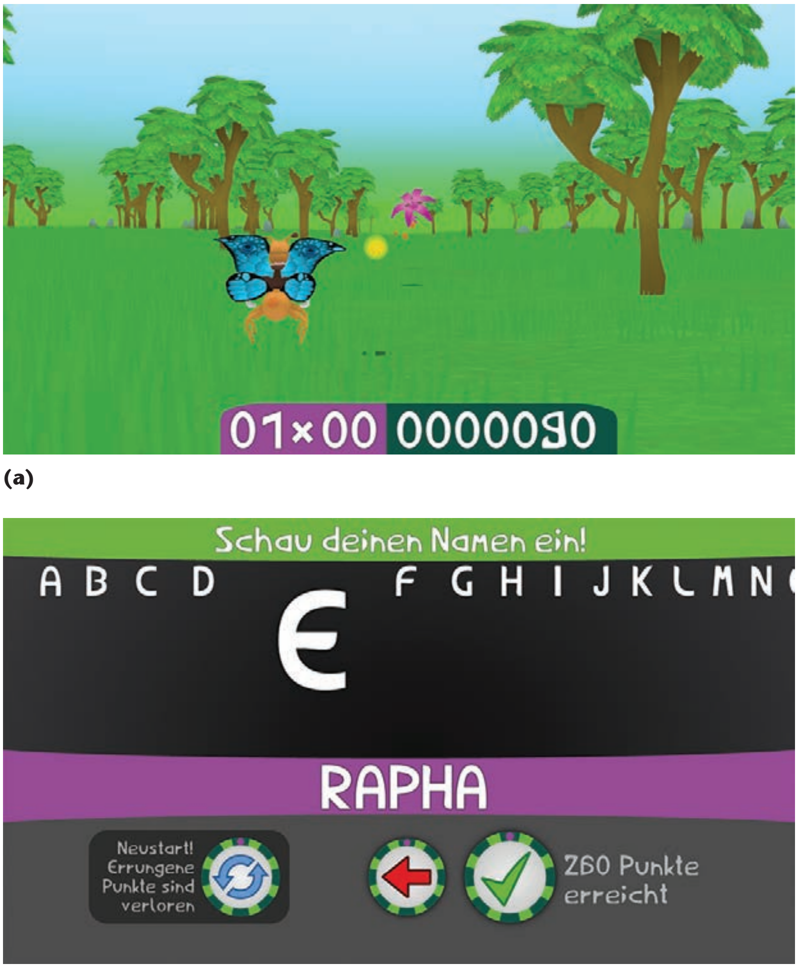
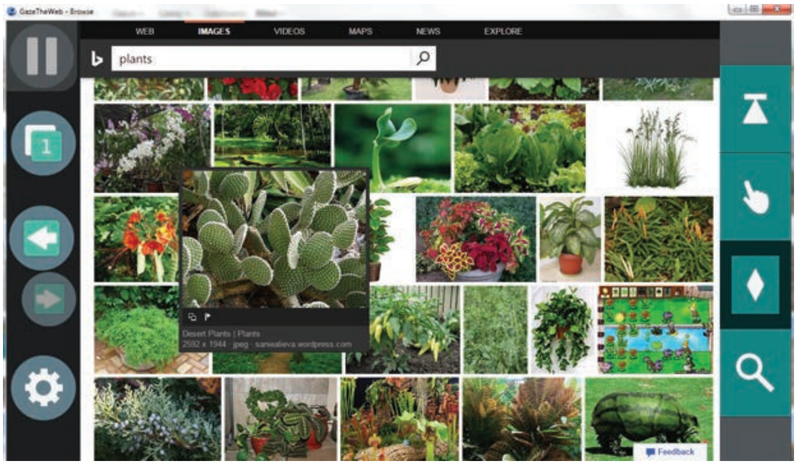


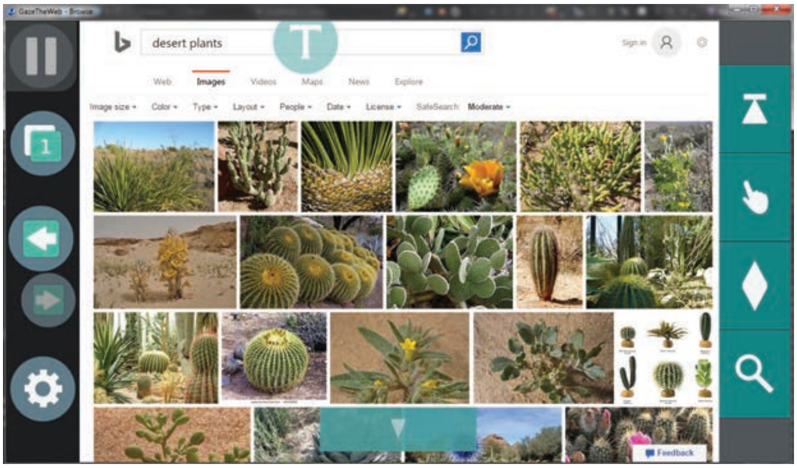
Figure 5. Eye-controlled gaming. The player (a) moves an animated object to select targets and (b) inserts a nickname for achieving a high score.

In this context, we have already conducted experiments and employed human fixation patterns to identify the most salient region of images, because defining a good crop requires a model that explicitly represents important image content. Our analysis of the Schau genau! game data implies that human fixations are very particular in identifying important image content. Figure 7 shows a sample image from the game data. The image on the far right is the visual saliency generated using eye fixations of players, which is more accurate compared to automatic saliency detection algorithms such as GBVS, Itti-Koch, and Signature saliency,[13] often used to generate automated crops of the picture.

Implicit gaze feedback is also relevant in personalizing the user experience—especially



(a)



(b)

Figure 6. Gathering implicit feedback to enhance search results: After searching for plant images, the user (a) interacts with the results and zooms in on a desert plant. Based on this implicit observation of the user's gaze, the GazeTheWeb system (b) would re-rank the results and present desert plants.

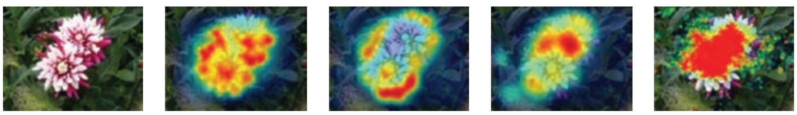


Figure 7. Cropping: Saliency patterns of image using automated algorithms and human gaze pattern: the (a) original image and images with (b) the GBVS, (c) Itti-Koch, and (d) Signature saliency overlays and with the (e) gaze overlay.

given that the sensory input performance largely depends on the individual's ability to process the cognitive information (for example, some users might be able to easily select the target with smaller dwell times compared to the time needed by others).

Integrating Additional Modalities

To overcome with the Midas-Touch problem, researchers in the HCI field have proposed different multimodal systems integrated with eye-gaze input. One kind of multimodal system combines the user's gaze with a hardware button—such as the MAGIC technique for pointing tasks.[2] This technique first warps the cursor to the vicinity of the target when the user stares at and wants to select it. Then the user can use a manual-pointing device to confirm the selection. Currently, we offer a similar functionality in GazeTheWeb by integrating the keyboard with gaze input. The user can look at the desired target in the screen and press a predefined hotkey (the Enter key) for the desired action. Another kind of multimodal system fuses the gaze and speech.[7]

For people with motor disabilities who can't use an external physical input device, integrating other psycho-physiological signals seems more appropriate. Furthermore, eye signals attentively reflect brain activities—that is, where users are looking indicates what they're processing in their minds, and how long they're looking at something indicates how much processing effort is needed (one example is the eye-mind hypothesis[14]). In this regard, electroencephalogram data from Brain Computer Interface (BCI) devices provide insight into how the brain works and helps us understand stress and other neural artifacts that can be incorporated to enhance gaze-based interaction.

Brain signals from alpha and beta frequency bands, together with eye-tracking signals, could provide more control in a multimedia interaction environment. With our MAMEM partners, we're currently investigating several fusion techniques to enhance the performance of eye-controlled interfaces via BCI and biosensors. We're examining the sensorimotor rhythms (SMRs) to understand signals that indicate state changes while the user navigates on the GazeTheWeb browser for a reading (inspection) or selection task. This "switching" task or state change task is being tested with SMRs in conjunction with signals from the eye-tracking system to eradicate or reduce the Midas-Touch problem. Moreover, SMRs could support complex multimedia interaction tasks like image editing (rotating images with motor imagery actions).

Furthermore, we examine error related potential (ErrP), which offers a natural way to detect errors for automatic error correction (AEC) with the EEG sensor. ErrPs have been used for AEC in BCIs (such as to correct misspellings in P300 or cVEP-based spellers[15]) but not in combination with eye-controlled interfaces. At first, our focus will be to correct misspellings, which will be activated when an ErrP signal is detected. In the future, we'll use ErrP on the more complex error scenarios of multimedia interaction. **MM**

Acknowledgements

This work is part of project MAMEM, which has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 644780.

References

1. T. van Gog and K. Scheiter, "Eye Tracking as a Tool to Study and Enhance Multimedia Learning," *Learning and Instruction*, vol. 20, no. 2, 2010, pp. 95–99.
2. P. Majaranta and A. Bulling, "Eye Tracking and Eye-Based Human–Computer Interaction," *Advances in Physiological Computing*, Springer, 2014, pp. 39–65.
3. R. Biedert et al., "The Text 2.0 Framework: Writing Web-Based Gaze-Controlled Realtime Applications Quickly and Easily," *Proc. 2010 Workshop on Eye Gaze in Intelligent Human Machine Interaction*, 2010, pp. 114–117.
4. T.C. Walber, A. Scherp, and S. Staab, "Smart Photo Selection: Interpret Gaze as Personal Interest," *Proc. 32nd Ann. ACM Conf. Human Factors in Computing Systems*, 2014, pp. 2065–2074.
5. D. Miniotas, O. Špakov, and I.S. MacKenzie, "Eye Gaze Interaction with Expanding Targets," *CHI 04 Extended Abstracts on Human Factors in Computing Systems*, 2004, pp. 1255–1258.
6. O. Špakov, "Comparison of Eye Movement Filters Used in HCI," *Proc. Symp. Eye Tracking Research and Applications*, 2012, pp. 281–284.
7. M. Kumar et al., "Improving the Accuracy of Gaze Input," *Proc. 2008 Symp. Tracking Research & Applications*, 2007, pp. 65–58.
8. C. Lankford, "Effective Eye-Gaze Input into Windows," *Proc. 2000 Symp. Eye Tracking Research & Applications*, 2000, pp. 23–27.
9. P. Majaranta et al., "Auditory and Visual Feedback During Eye Typing," *CHI 03 Extended Abstracts on Human Factors in Computing Systems*, 2003, pp. 766–767.
10. C. Schaefer et al., "Schau genau!—An Eye Tracking Game with a Purpose," *Workshop on the Applications for Gaze in Games at CHI Play*, 2014.
11. R. Menges et al., "EyeGUI: A Novel Framework for Eye-Controlled User Interfaces," to be published in *Proc. 9th Nordic Conference on Human-Computer Interaction* (NordiCHI), 2016.
12. O. Oyekoya and F. Stentiford, "Perceptual Image Retrieval Using Eye Movements," *Int'l J. Computer Mathematics*, vol. 84, no. 9, 2007, pp. 1379–1391.
13. A. Borji and L. Itti, "State-of-the-Art in Visual Attention Modeling," *IEEE Trans Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, 2013, pp. 185–207.
14. M.A. Just and P.A. Carpenter, "Eye Fixations and Cognitive Processes," *Cognitive Psychology*, vol. 8, 1976, pp. 441–480.
15. M. Spüler, W. Rosenstiel, and M. Bogdan, "Online Adaptation of a c-VEP Brain-Computer Interface (BCI) Based on Error-Related Potentials and Unsupervised Learning," *PLoS ONE*, vol. 7, no. 12, 2012, pp. 1–11.

Chandan Kumar is a postdoctoral researcher at the Institute for Web Science & Technologies (WeST), Universität Koblenz-Landau, Germany. Contact him at kumar@uni-koblenz.de.

Raphael Menges is a student research assistant at the Institute for Web Science & Technologies (WeST), Universität Koblenz-Landau, Germany. Contact him at raphaelmenges@uni-koblenz.de.

Steffen Staab is a professor for databases and information systems, and head of the Institute for Web Science & Technologies (WeST), Universität Koblenz-Landau, Germany. He also holds a chair for Web and Computer Science with the Web and Internet Science Research Group (WAIS) at the University of Southampton, UK. Contact him at staab@uni-koblenz.de.

This article originally appeared in IEEE MultiMedia *vol. 23, no. 4, 2016.*

The Future of Big Data Is … JavaScript?

Jimmy Lin and Kareem El Gebaly • *University of Waterloo*

The future of Big Data is … JavaScript? No, seriously, hear us out.

JavaScript has already made significant inroads as an integrated technology stack for building user-facing applications, both on the front end (JavaScript running in the browser) and the back end (JavaScript running in Node.js). In this article, we explore the idea of using JavaScript in Big Data platforms, both on the front end (for example, how data scientists interact with analytical tools) and on the back end (such as distributed execution infrastructure). Could the future be … JavaScript everywhere?

Given the anticipated wince-inducing reaction to this question from any sane developer, let's get this out of the way first:

JavaScript is a terrible language.

JavaScript is terrible in so many ways. Instead of recounting them all, we'll just refer to "WAT," Gary Bernhardt's incredibly funny talk.[1] Nevertheless, we should cut Brendan Eich (the creator of JavaScript) some slack, considering that he developed the language in 10 days. We're sure neither he nor anyone else expected JavaScript to become so successful and ubiquitous. Anywhere there's a browser, there's (almost always) JavaScript. Your connected refrigerator or toaster in the near future? It might have a browser for its user interface, which means it'll likely run JavaScript.

But we're getting ahead of ourselves. Let's start with a quick recap of computing history to see how we got here.

A Brief History

In the mid-1990s, Sun Microsystems (gobbled up by Oracle in 2010) came up with a brilliant slogan, "write once, run anywhere," to describe the promises of Java in offering seamless cross-platform development. In the early days of the Web, one of the main selling points of Java was its ability to run inside the browser via applets — so you can push server-side code over to the client with minimal hassle. It sounded like a great idea, except that it didn't actually work. Java applets were clunky, slow, and insecure. Eventually, the world relegated them to the dustbin of computing history — at best a historical curiosity today.

In the end, of course, Java did just fine. Although today hipster languages such as Scala and Clojure jockey for attention, the broader ecosystem of the Java Virtual Machine (JVM) is well-entrenched on the server. Thanks to Hadoop, Spark, Flink, and other Big Data platforms, the JVM has become the de facto execution environment for Big Data applications. On the client side, Java runs on more than a billion devices around the world via Android. Even though it didn't deliver on the "write once, run anywhere" promise, by any standard Java is a smashing success.

Despite the similarity in names, which was largely a marketing ploy, JavaScript is entirely unrelated to Java (except that both share C-like syntax). It was originally designed as a lightweight in-browser interpreted language for non-professional programmers, but over the years JavaScript has become the dominant language for Web development, practically

 2469-7087/17/$33.00 © 2017 IEEE

Big Data Bites

"Big Data Bites" is a regular department in *IEEE Internet Computing* that aims to deliver thought-provoking and potentially controversial ideas about all aspects of Big Data. Interested in contributing? Drop me a line!

—*Jimmy Lin*

unchallenged in its hegemony. In fact, front-end development today is synonymous with HTML, CSS, and JavaScript. In the beginning, JavaScript was limited to those dumb "see, I can compute factorial in the browser" and "oh look, I can change the background color of the page with a button click" scripts. Today, Web apps written in JavaScript offer user experiences rivaling that of native desktop apps. For example, Gmail is comprised of more than 400,000 lines of JavaScript (as of 2010).[2]

On the client side, JavaScript comes much closer to delivering the promise of "write once, run anywhere" than Java ever did — evidenced by the fact that JavaScript-rich Web apps "just work" nearly all the time. They work on Intel chips. They work on ARM chips. They work on Windows, Linux, Mac, and other operating systems. They work on Firefox, Chrome, Safari, and other browsers. They work on desktops, laptops, tablets, and mobile phones. Sure, cross-platform Web development remains a herculean effort with endless shims and special handling of weird incompatibilities; but for the average end user, the experience is fairly seamless. That's a pretty remarkable feat.

The JavaScript Server Incursion

And so for a while, we lived in a stable bipolar world. JavaScript reigned in the browser on the client side, and Java held a dominant position on the server (although with competition from PHP, Rails, Python, and other stacks). REST APIs served as platform- and language-agnostic connections between front ends and back ends. All was well.

Then, some developers had to upset this balance by exploring the question, "What if we started running JavaScript on the server also?" Server-side JavaScript is an idea that dates back to 1995, soon after JavaScript appeared in the browser, but no one took it seriously until the mid-2000s. Developers started exploring the idea for a variety of reasons: One of the most compelling is the reduction of impedance mismatch between front- and back-end development, particularly for building rich, interactive Web apps. Some of the challenges were technical — the friction that comes from switching between different languages and environments when the end goal is to deliver a seamless user experience,[3] but some were organizational — writing everything in JavaScript breaks down team structures that segmented developers into front- and back-end teams, with all the associated communications barriers of silos.[4]

All of this became practical with the advent of high-performance JavaScript engines, most notably Google's open source V8 (https://developers.google.com/v8/). Everything came together with Node.js, the popular open source runtime environment for server-side JavaScript.

There have been a number of high-profile success stories of Node.js. In 2012, LinkedIn mobile moved from running Ruby on Rails on the back end to Node.js and was able to replace 30 servers with just three to serve the same traffic.[5] Netflix, PayPal, Uber, and many others are all big users of Node.js. Development with Node.js meshes well with the "microservices" architecture in vogue today, and the advent of Docker (www.docker.com) simplifies many aspects of deployment. All of these technologies fit together nicely to offer an integrated stack for building user-facing applications.

Before moving on, there's another historical footnote worth mentioning. Beyond applets, there has been another concerted attempt at pushing Java onto the client. Google Web Toolkit (GWT; www.gwtproject.org), initially released in 2006, is a framework that lets users develop Web apps all in Java. It powered the ill-fated Google Wave, if anyone remembers that. GWT is, for all intents and purposes, dead. Somewhat ironically, GWT cross-compiles Java into JavaScript that's deployed on the front end, so this attempted incursion by Java into the browser is perhaps best described as a Trojan Horse.

JavaScript and Big Data

So far, we've (hopefully) established that for user-facing applications, adopting a pure JavaScript stack (both on the front end and back end) is a fairly compelling choice. Let's now examine the radical proposition of this article — that there are similarly compelling reasons to adopt JavaScript for Big Data applications, both on the front end and back end. In contrast to user-facing applications, Big Data applications are internal-facing, used mostly by an organization's data scientists (and maybe higher-level pointy-hair types via dashboards). To be clear: by "front end," we mean the tools that data scientists use to access analytical capabilities over large datasets, and by "back end," we mean the distributed processing infrastructure (for example, Hadoop and Spark) that provides those capabilities.

What's the front end for data scientists? Traditionally, it's just been a shell — in the old days, this might have been Pig's "Grunt" shell or the Hive shell; but more recently, it's likely to be the Spark shell. Good old command-line interfaces!

However, changes are brewing. Browser-based notebooks (such as Jupyter; http://jupyter.org) have gained tremendous popularity with data scientists in recent years, for a variety of reasons: The tight integration of code and execution output elevates the analytical process and its products to first-class citizens, because the notebook itself can be serialized, reloaded, and shared. The ability to manipulate, rearrange, and insert snippets of code (in "cells") lines up well with the iterative nature of data science and a wide range of analytics tasks. With the seamless integration of browser-based notebooks and scalable data analytics platforms, code written in a browser-based notebook can be executed on a cluster and the results can be further manipulated in the notebook. The commercial Databricks platform represents a big bet that browser-based Big Data analytics will be the way of the future.

This means that the browser may become the shell. And remember, JavaScript reigns in the browser.

Furthermore, the end product of data science is often a visualization to tell a story. In many cases, these visualizations are interactive, giving the data scientist (and her colleagues, managers, clients, and so on) a chance to twiddle knobs and drag sliders in exploring the data. In other words, the "output" of data science is actually a custom-built Web app that encapsulates the analytical results! The preferred visualization tool these days is D3.js (https://d3js.org). Guess what — that's JavaScript, too. In other words, ultimately Big Data is user-facing also — with just a different set of users.

Currently, we have a "canonical" Big Data analytics pipeline that looks something like this: Spark clusters running in the datacenter (the back end) talking to a Spark client (either Scala, Python, or R) that might be running in a browser (notebooks), connected somehow to a Web app (an interactive D3.js visualization) that's the final analytical product. The connection between Spark clusters and Scala/Python/R is fairly well developed, but interfacing with D3.js today is likely accomplished via JavaScript Object Notation (JSON) or comma-separated value (CSV) dumps, which is clumsy. Why not cut out the middleman and just use JavaScript?

Let's try to refine this argument by examining the modern data science workflow in more detail, leaving aside the final visualization step for now. It's well known that the work of data scientists, particularly in cases of exploratory tasks, is highly iterative, where they "poke" at the data from many different angles. Frequently, these tasks focus on a small subset of the data — for example, the data warehouse stores six months of log data, but the data scientist is only interested in data from the last week. Here are some ways she can go about accomplishing her task:

1. She repeatedly queries the entire dataset, but filters for the subset of interest. Achieving low query latency is entirely dependent on the analytical engine properly optimizing away unneeded work, which is dependent on the data's physical storage (how it's partitioned, compressed, and so forth).
2. She materializes the data of interest. In an analytical database, this typically involves selecting into a temporary table or creating a materialized view; in Spark, this typically involves writing intermediate results onto the Hadoop Distributed File System (HDFS). Subsequent queries would then be posed against this smaller dataset.
3. She materializes the data of interest and then copies the data locally (to her laptop, for example) for further manipulation. This might involve running Spark locally or dumping data into a local relational database and then issuing additional queries.

There are issues with all three approaches. In the first, the analytics engine might not be smart enough to efficiently optimize the query, or there might be inherent limits to query execution speed. As a simple example, if the dataset is hash-partitioned, pulling recent records might still involve distributed scans with large fan-outs. After standard optimizations such as columnar layouts and predicate pushdowns, network latency and coordination overhead start becoming the bottlenecks.

In the second and third scenarios, queries over the materialized data might no longer run efficiently because the startup cost of the analytics engine (such as Spark) dominates the actual processing time. For some datasets, it might be easier to wrangle on the command line using `sed`, `awk`, and so on. By definition, efficient queries over small datasets isn't a scenario for which Big Data processing platforms optimize.

In the second scenario, materializing temporary data creates a different set of challenges. In analytical databases, creating materialized views might require elevated privileges (not granted to everyone), and the ability to create temporary tables presents a different set of data management headaches. For example, who (data scientists themselves?) or what (automated processes?) "cleans up" 51 tables named variants of tmp or test? The same goes for Spark: go look in your HDFS home directory right now and count up all the directories named test1, tmp42, and so forth. Now try to remember how those intermediate results were generated.

The third scenario has the downside of an awkward workflow. For a relational database, this involves the data scientist copying data over to her laptop and ingesting the data locally before additional queries can be run. Furthermore, this approach essentially requires maintaining two separate analytics stacks (one on the client and one on the server). Say, she was playing with a new unstable branch of Spark on her laptop to test out a new feature (which hasn't been rolled out to production yet), but in the new branch, the API has changed slightly. What about the custom machine-learning library that her organization deploys on top of Spark — is the version running on the cluster the same as the one running on her laptop?

Now imagine an alternative: What if we had something like Spark.js, where we could transparently write Spark code in JavaScript? Fundamentally, this isn't any different from PySpark, which is based on Py4J (www.py4j.org), a bridge that enables Python programs running in a Python interpreter to dynamically access Java objects in a JVM. Similarly, Spark.js could be written with the help of Rhino (www.mozilla.org/rhino), an open source implementation of JavaScript written entirely in Java.

In the current Spark API, "collect" and related methods materialize a resilient distributed dataset (RDD), the results of which are sent over to the client. In Spark.js, JVM objects could be transparently converted into native JavaScript objects, ready for subsequent manipulation and seamless integration with D3.js for visualization. All of this would run in the browser (any standards-compliant one, on any platform) or alternatively Node.js.

Wait, it gets even better. At this point, Node.js already runs your user-facing applications, right? Remember that complex extract, transform, and load (ETL) pipeline for pumping log data into Kafka (http://kafka.apache.org) and eventually HDFS? All that custom code for data cleaning, schema transformation, and so on can now be written in JavaScript. Developers often write log messages in JSON these days, right? Note what the "J" in JSON stands for! So, with everything in JavaScript, you can retain the convenience of what you were already doing for user-facing applications but gain seamless integration with the data collection and analytics stack. This is a win from the friction-reduction perspective and potentially the performance perspective as well — ETL pipelines in JavaScript might reduce the amount of unnecessary data transformation, thus decreasing latency and increasing throughput. Although textual JSON isn't particularly efficient as a storage format (even when compressed), the physical layout could be easily optimized using Parquet (http://parquet.io), which is essentially built around a JSON data model — thus reaping the benefits of columnar layouts. You can have your cake and eat it, too!

One final leap: Note that the actual execution of Big Data analytics still occurs on Spark, inside the JVM. Spark.js merely provides convenient JavaScript integration. Why don't we "go all the way" and replace Spark completely with a native JavaScript analytics engine? We wonder, what would a Big Data analytics platform written in the style of microservices look like? This is perhaps not an outrageous idea, because such a platform would be the logical endpoint of a "JavaScript everywhere" vision.

A Small Step

To examine the feasibility of this "JavaScript everywhere" vision, we recently implemented a small prototype analytical RDBMS called *Afterburner* in JavaScript.[6] The entire system runs completely inside a browser with no external dependencies (or alternatively, in Node.js). Afterburner generates compiled query plans[7,8] that exploit two JavaScript features: typed arrays and asm.js, which we explain next.

Array objects in JavaScript can store elements of any type and aren't arrays in a traditional sense (compared to say, C), because consecutive elements might not be contiguous; furthermore, the array itself can dynamically grow and shrink. This flexibility makes the data structure accessible for novice programmers but limits optimizations that JavaScript engines can perform both during compilation and at runtime. In the evolution of JavaScript, it became clear that the language needed more efficient methods to quickly manipulate binary data: typed arrays are the answer.

In conjunction with typed arrays, Afterburner takes advantage of asm.js, a strictly typed subset of JavaScript that's designed to be easily optimizable by an execution engine. A key feature of asm.js is the use of type hints, which essentially introduces a static type system while retaining backwards compatibility with "vanilla" JavaScript.

Any JavaScript block of code can request validation as valid asm.js via a special prologue directive, which happens when the source code is loaded. Validated asm.js code is amenable to ahead-of-time (AOT) compilation, in contrast to just-in-time (JIT) compilation with vanilla JavaScript. Executable code generated by AOT compilers can be quite efficient, through the removal of runtime type checks, operations on unboxed (that is, primitive) types, and the removal of garbage collection.

At a high level, Afterburner translates SQL into the string representation of an asm.js module (that is, the physical query plan), calls `eval` on the code, which triggers AOT compilation and links the module to the calling JavaScript client, and finally executes the module. The typed array storing all the tables (essentially, the entire database) is passed into the module as a parameter and the results are returned by the module.

Before summarizing some preliminary results, it's worthwhile to highlight how far JavaScript has come as an efficient execution platform, from its much-maligned performance in the early days. Today, modern browsers embed powerful JavaScript engines capable of running real-time collaboration tools and online multiplayer games. One impressive feat is QuakeJS (www.quakejs.com), a port of the classic first-person shooter Quake 3, that *runs completely in the browser*. The original C code was compiled into asm.js using Emscripten (http://emscripten.org), which takes LLVM bitcode and emits JavaScript as its target output. The game runs at a good frame rate and is definitely playable. As a rough heuristic, C code compiled to asm.js (which, recall, is just JavaScript) using Emscripten runs at about half the speed of the C program running natively (directly on Linux, for example). For many applications, the tradeoff of performance for ubiquitous execution inside a browser is worthwhile.

We evaluated Afterburner against the popular open source analytical database MonetDB,[9] using data from the TPC-H benchmark. At a scaling factor of 1 gigabyte, which yields a lineitem table with 6 million rows and an orders table with 1.5 million rows, experiments with a few simple queries show that our prototype achieves comparable performance to MonetDB running natively on the same machine. See our report for additional details.[6] Of course, we're quick to qualify that Afterburner is a prototype with limited functionality, but our evaluations affirm at least the *feasibility* of in-browser analytics using JavaScript.

From Split Execution to the Connect Refrigerator

Recall that one of the advantages of JavaScript everywhere is eliminating the distinction between front- and back-end code; see, for example, discussion by Netflix.[3] Returning to our data scientist, she can perform some analytics tasks on the cluster, bring intermediate data into her browser, and continue working without interruption. Front-end code can be pushed to the back end, and back-end code can be brought to the front end, all seamlessly.

In such a scenario, the data scientist explicitly decides which parts of her queries run where — the next logical step would be for a query optimizer to figure that out for her automatically. In fact, researchers have already explored this idea in the 1990s;[10] in some cases, the placement of query operators on the client can lead to better performance, and obviously, reduces load on the back end. JavaScript brings a fresh take on this decades-old idea of splitting query execution across the client and server: one of the challenges with this thread of previous work is the proper encapsulation of the execution context to facilitate operator placement. With JavaScript, this is substantially simplified.

A major selling point of JavaScript is its ubiquity: everywhere there's a browser, there's JavaScript. The browser might even be on a phone: we've gotten Afterburner to run on a mobile phone, just to show it's possible. One not-too-unrealistic scenario might be to run something like Afterburner on a tablet to manipulate data: spreadsheets already run on tablets today — how would this be substantially different? Imagine running a Spark job from the browser inside your tablet to pull some data from the cluster, getting on an airplane, and continuing to explore the data while completely offline. Today, you could do this by exporting data into a spreadsheet app on the tablet, which is just a variant of the aforementioned third scenario, with all of its awkwardness.

If you've accepted our story so far, then it's not a particularly big leap to imagine extending the "JavaScript everywhere" vision to the Internet of Things (IoT) future: for example, check out Node-RED (http://nodered.org). In the not-so-distant future, imagine returning to your fully connected house, asking yourself, "Why is it so hot on the second floor?" and issuing the following query from the browser console on your fridge:

```
SELECT room, temp FROM sensors
WHERE floor = 2 and time = NOW;
```

This translates into a distributed query plan that gets pushed to all the sensors around the house, the results of which are then gathered back at the fridge and rendered as a heatmap — all in JavaScript.

Don't laugh — something like this has already been developed, built, and deployed in the context of distributed sensor networks: it's called TinyDB.[11] The only difference here is that we propose to implement everything in JavaScript.

Consider another example: you're at the grocery store and wonder if you need to buy milk. Pull out your phone and ask your fridge:

```
SELECT COUNT(*) FROM fridge
WHERE item = "milk";
```

Of course, actual SQL queries are likely to be hidden behind a snazzy UI within an app (or connected via intelligent agents such as Siri or Cortana), but behind the scenes, it could all be running JavaScript.

In fact, this vision of the IoT has one significant advantage over current trends — it can potentially serve as a counterweight to the desire of companies to centralize everything. Nearly all IoT gadgets today, including personal fitness trackers, pipe all data streams into the cloud, and then offer users an interface (or API) to access their own data (which of course the companies can change at any time). In other words, they control all your personal data.

Just how creepy is this? Samung's SmartTV comes with voice recognition that allows you to issue verbal commands to it. It's a cute trick, but comes with a warning to watch what you say in front of your TV:

> *Please be aware that if your spoken words include personal or other sensitive information, that information will be among the data captured and transmitted to a third party.*[12]

Seriously? This is, literally, straight out of *1984*.[13]

One major advantage of the "JavaScript everywhere" vision is that, by erasing the distinction between front- and back-end code, we potentially lower the bar (by decreasing the complexity) of building distributed solutions, which will help counteract the dangers of centralization. In other words, JavaScript can help re-decentralize the Web.[14] Of course, we aren't so naive to think that companies will easily give up their data collection efforts (which lie at the heart of their business models), but the ability to seamlessly push queries down to physical devices provides a workable mechanism that might give users greater control over what is and isn't transmitted to the cloud.

Generalizing a bit, we could envision an entirely different class of "verb-centric" APIs. Almost all APIs today are "noun-centric," in that they hardcode the "verbs" in specific endpoints. For example, consider the common CRUD (create, read, update, delete) operations provided by many REST APIs: to invoke the API, the client supplies the "subject" (and sometimes the "object") while the "verbs" are fixed (that is, hardcoded in the API endpoint itself). What if, instead, the API fixed the subject and object while letting the client specify the verb? That's exactly what the aforementioned fridge query is doing — the client says, I know the data model that's stored in a distributed fashion on the device (the crucial distinction here, as opposed to a centralized data warehouse) — here's the verb, expressed as an SQL query that I want to run on the data. Give me back the results.

In fact, we've already been doing this for at least a decade — we're just describing MapReduce! A verb-centric API is just another way of saying "move the processing to the data." We ship the verb (that is, the mappers and reducers) over to the cluster, where the code executes. Spark takes it to the next level by providing an interactive shell that allows closures (the verbs) to be dynamically serialized, shipped over to the cluster, and executed in a distributed manner. JavaScript makes this possible everywhere. We can just reuse existing infrastructure — it might offend the sensibilities of many developers, but what's wrong with code shipping in textual form? It's already being done billions of times per day — JavaScript code on the Web is distributed as plain text! What's the alternative? Shipping VM images? Shipping Docker images? Both seem too heavyweight for widespread, democratized use.

Yes, there are many issues with verb-centric APIs that we're mostly sweeping under the rug, but we can respond briefly to a few obvious ones. What about security if a client is shipping verbs? First, we can restrict clients to trusted sources — you're probably not going to run an injection attack on your own fridge, in which case, perimeter defenses are a good start (a reasonable first-order approximation of how Hadoop security is handled these days). Second, this issue isn't very different from JavaScript injection attacks, something that's already well studied and continues to be actively researched. What about the exfiltration of potentially sensitive results from running the code? Leaving aside the "friendly clients" defense, this is a hot topic in security and privacy these days, which means that lots of smart people are thinking about it. Tying the challenges of JavaScript everywhere to existing research problems greatly increases the chances that we (collectively) will come up with workable solutions.

In many ways, JavaScript is like the hobbits of Middle Earth: both have been criticized to be slow, clumsy, and weak. They're also known to be very friendly (JavaScript is easy to learn). They live away from danger at the edge of the realms in lovely dwellings (D3.js visualizations), far from the troubles at the center of the action (the back end). However, when tested (Node.js), they have proven to be worthy. JavaScript, instead of successfully destroying the One Ring, might be successful in wielding it. Front ends, back ends, applications, and analytics: JavaScript will, in the light, bind them.

Acknowledgments

We thank Ashraf Aboulnaga and Todd Lipcon for comments on earlier drafts of this article.

References

1. G. Bernhardt, "WAT," *Destroy All Software*, software screencast, 2012; www.destroyallsoftware.com/talks/wat.
2. A. de Boor, "Gmail: Past, Present, and Future," *Proc. Conf. Web Application Development*, 2010; www.usenix.org/legacy/event/webapps10/tech/slides/deboor.pdf.
3. K. Baxter, *Making Netflix.com Faster*, blog, 2015; http://techblog.netflix.com/2015/08/making-netflixcom-faster.html.
4. J. Harrell, *Node.js at PayPal*, blog, 2013; www.paypal-engineering.com/2013/11/22/node-js-at-paypal.
5. R. Paul, "A Behind-the-Scenes Look at LinkedIn's Mobile Engineering," *Ars Technica*, 2 Oct. 2012; http://arstechnica.com/information-technology/2012/10/a-behind-the-scenes-look-at-linkedins-mobile-engineering.
6. K. El Gebaly and J. Lin, "Afterburner: The Case for In-Browser Analytics," 2016, https://arxiv.org/pdf/1605.04035.
7. K. Krikellas, S. Viglas, and M. Cintra, "Generating Code for Holistic Query Evaluation," *Proc. 26th Int'l Conf. Data Eng.*, 2010, pp. 613–624.
8. T. Neumann, "Efficiently Compiling Efficient Query Plans for Modern Hardware," *Proc. Very Large Data Bases*, vol. 4, no. 9, 2011, pp. 539–550.
9. P.A. Boncz, M. Zukowski, and N. Nes, "MonetDB/X100: Hyper-Pipelining Query Execution," *Proc. 2nd Biennial Conf. Innovative Data Systems Research*, 2005, pp. 225–237.

10. M.J. Franklin, B.T. Jonsson, and D. Kossmann, “Performance Tradeoffs for Client-Server Query Processing,” *Proc. 1996 ACM Sigmod Int’l Conf. Management of Data*, 1996, pp. 149–160.
11. S.R. Madden, M.J. Franklin, J.M. Hellerstein, and W. Hong, “TinyDB: An Acquisitional Query Processing System for Sensor Networks,” *ACM Trans. Database Systems*, vol. 30, no. 1, 2005, pp. 122–173.
12. N. Lomas, “Today in Creepy Privacy Policies, Samsung’s Eavesdropping TV,” *Tech Crunch*, 2015; http://techcrunch.com/2015/02/08/telescreen.
13. P. Higgins, Twitter post, 8 Feb. 2015, 1:35 a.m.; https://twitter.com/xor/status/564356757007261696.
14. E. Mansour et al., “A Demonstration of the Solid Platform for Social Web Applications,” *Proc. 25th Int’l Conf. Companion on World Wide Web*, 2016, pp. 223–226.

Jimmy Lin holds the David R. Cheriton Chair in the David R. Cheriton School of Computer Science at the University of Waterloo. His research lies at the intersection of information retrieval and natural language processing, with a particular focus on Big Data and large-scale distributed infrastructure for text processing. Lin has a PhD in electrical engineering and computer science from MIT. Contact him at jimmylin@uwaterloo.ca.

Kareem El Gebaly is a PhD student in the David R. Cheriton School of Computer Science at the University of Waterloo. His research interests include data management and self-managing data analytics. El Gebaly has an MMath in computer science from the University of Waterloo. Contact him at kareem.elgebaly@uwaterloo.ca.

This article originally appeared in IEEE Internet Computing *vol. 20, no. 5, 2016.*

Editor:
Theresa-Marie Rhyne

Data-Driven Healthcare: Challenges and Opportunities for Interactive Visualization

David Gotz and David Borland
University of North Carolina at Chapel Hill

The adoption and use of health information technology is increasing dramatically around the world. In the United States, the Health Information and Technology for Economic and Clinical Health (HITECH) Act was enacted in 2009 to promote the adoption and use of health IT, including specific incentives to accelerate the adoption of electronic health record (EHR) systems by healthcare providers. As of 2014, 76 percent of US hospitals had adopted an EHR system, representing a more than eightfold increase from the 9 percent adoption rate in 2008.[1] Outpatient facilities have also seen a similarly dramatic rise in EHR adoption rates. Similar trends can be seen elsewhere around the world, with many counties well ahead of the United States in health IT adoption.[2] The long-term use of these EHR systems is enabling the construction of enormous collections of detailed longitudinal medical data, containing a vast array of information about large and varied populations of patients.

The industry's widespread digitization efforts, along with changing business models that are incentivizing more efficient and effective care delivery, are reshaping one of the largest sectors of the world's economy. Beyond basic benefits (such as improved sharing of medical information and a reduction in duplicate tests or procedures), many have recognized the "inevitable" application of big data[3] resources to enable data-driven, learning health systems.[4] Such systems promise to use ever-improving data-driven evidence to help doctors make more precise diagnoses, institutions identify at-risk patients for intervention, clinicians develop more personalized treatment plans, and researchers better understand medical outcomes within complex patient populations.

Given the scale and complexity of the data required to achieve these goals, along with the domain expertise and analytical rigor demanded by the use cases outlined here, advanced data visualization tools have the potential to play a critical role. In particular, effective visualization technologies have the potential to transform raw data and the outputs of complex computational models into actionable insights that improve patient care, enable more effective population management, and support advanced research to better understand health outcomes and treatment efficacy.

The use of visualization within the healthcare domain has a long and storied history (see Figure 1). However, there are new and unique challenges emerging in today's data-rich healthcare industry where modern interactive visualization methods can play a critical role. The enormous potential of these techniques is reflected in several recent developments. For instance, recent articles within the visualization literature have provided surveys of emerging research targeting specific healthcare-related research problems.[5,6] In addition, the American Medical Informatics Association (AMIA) established in 2015 a formal Visual Analytics Working Group to foster research and development in this high-priority area. Similarly, the *Journal of the American Medical Informatics Association* published a special issue in early 2015 dedicated to visual analytics in healthcare.[7] That issue included articles about new research as well as a systematic review of the state of the art in the field.[8] These efforts, which are building a bridge between the medical and visualization communities, are also reflected in the recent Visual Analytics in Healthcare (VAHC) workshops, held annually each fall since 2010 at either the AMIA's Annual Symposium or the IEEE VIS conference.

Data-related problems in healthcare are similar in many ways to those in other domains. Challenges of data integration, wrangling, ease of use, and interpretability are all central issues. However, the healthcare discipline also introduces a number of domain-specific challenges:

- *breadth of use*, from individualized point-of-care to large-scale population health applications;
- *data complexity*, including large numbers of patients, large numbers of heterogeneous variables, data linking across multiple sources, and missing or incomplete data; and
- *statistical rigor*, where "interesting" is not sufficient given the life-or-death stakes within the healthcare domain.

By addressing these challenges and integrating the existing workflows of healthcare practitioners, interactive data visualization has the potential to become a useful, and perhaps essential, tool for a modern data-driven healthcare ecosystem.

Breadth of Use

In future healthcare systems, a broad range of practitioners must derive insights from large collections of data to make evidence-based discoveries and/or decisions. From a doctor making individual patient treatment decisions at the point of care to population health researchers attempting to understand the spread of disease, to be most effective, visualization tools must be tailored to the unique workflows of each type of practitioner. For this reason, it is essential for visualization researchers to collaborate closely with their intended users, providing additional opportunities to support and incorporate domain-specific practices and techniques.

Patient-Centered Point-of-Care Applications

Patient-centered point-of-care applications focus on providing support for information communication and analysis for a single patient. Such visualizations must provide clinicians (doctors, nurses, social workers, care coordinators, and so on.) with the information required to more effectively and efficiently render service. Even within this relatively narrow focus area, the use cases can be broad and the amount of data can be overwhelming. Clinicians may need to synthesize data across years of a patient's longitudinal medical record to understand the evolution of a given medical condition. This data can include hundreds of encounters and thousands of medical events (such as diagnoses, procedures, medications, and lab tests)

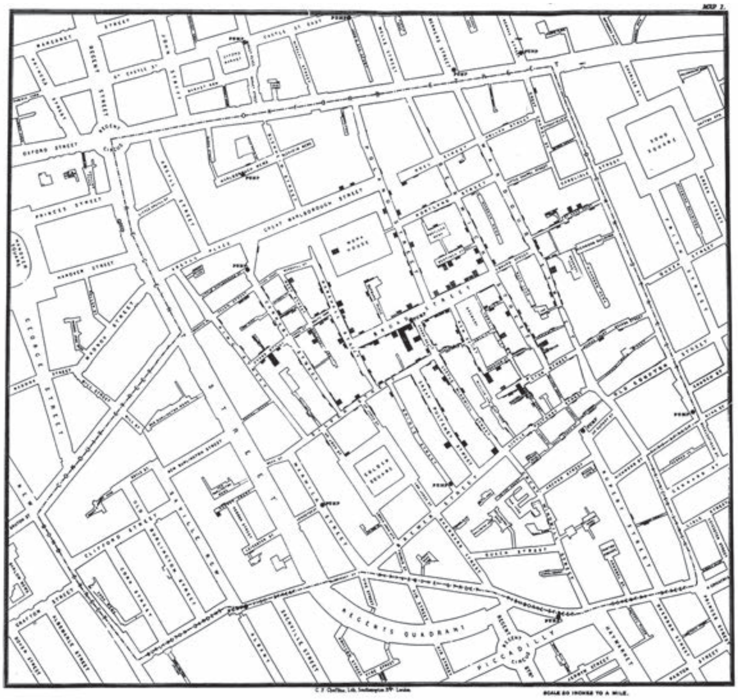


Figure 1. An early use of visualization within the healthcare domain. John Snow's map showing clusters of cholera cases during an outbreak in 1854 London is often cited as a seminal moment in the field of epidemiology.

and unstructured clinical notes. Recent advancements in genomic sequencing and cataloging genetic markers make the data even more complex.

Summarizing this information and providing the most relevant view of the data given the current clinical context is therefore essential. However, the appropriate visual summary is context-specific. The most effective view of a patient's medical record during a cardiology visit may differ significantly from the view required during a visit with an oncologist. This challenge, therefore, is one that requires advancements in both visualization and the associated analytics required to support patient-centered organization, prioritization, and summarization of a patient's medical record.

Moreover, representing an individual patient's data in isolation is not always sufficient. In many cases, the promise of precision medicine—the customization of care such that medical decisions are tailored to individual patients based on data, as a supplement to traditionally used peer-reviewed research and generalized clinical guidelines—requires that a patient's personal data be contextualized with respect to aggregate data from other patients. For example, it might be useful to view treatment options for a given patient in the context of outcomes that resulted from the various treatments received by similar patients.[9] This can require both sophisticated analytics (for example, to determine clinically relevant similarity) as well as visualizations designed to help clinicians quickly but rigorously compare aggregate measures computed across multiple heterogeneous groups of patients.

Even basic point-of-care activities, such as reconciling medication and shift changes in hospital settings, can be information-intensive activities for which visualization can play a crucial role. While requiring less-sophisticated analytics, visualization tools that improve the accuracy and reliability of these tasks have the potential to greatly impact health outcomes.

The usability of a proposed visualization design is also critical for all point-of-care applications. To be successful, any data-driven software tool must support reliable, efficient, and statistically sound decisions by its user population. In point-of-care settings, the intended users may have little experience with complicated interactive visualization systems. With the increased adoption of tablets and mobile devices as tools for healthcare providers, touch interfaces with limited screen space must also be considered. Appropriate human-computer interaction (HCI) guidelines should be followed, with a focus on simple and easily interpretable designs. Usability testing must be employed to ensure that the system enhances, and does not hinder, a user's ability to provide quality care.

Patient-Facing Applications

Patient-facing applications have requirements that are in many ways similar to the point-of-care applications just discussed. Patients, like clinicians, require tools that enable personalized and contextualized communication of medical histories and treatment alternatives. However, visual interfaces that target patients (or their caregivers) must be tailored for a lay audience with potentially limited numeracy and medical knowledge. Therefore, simple straightforward designs with recognized graphical representations are likely to be most successful.

Despite these constraints, engaging designs that can improve patient involvement in the medical process should be explored. Visualization tools that can increase patient engagement have the potential to improve treatment adherence rates and associated patient outcomes. Moreover, visualization tools have the potential to support physician-patient communication, including via storytelling techniques.

Population Management Applications

Population management applications, on the other hand, focus on supporting institutional policymakers with the design of population-based interventions. These use cases are of growing importance in healthcare systems, such as in the US where the industry is shifting from fee-for-service to capitation-based models that incentivize the intelligent allocation of resources to patients most in need.

To support these emerging models, visualization tools must be developed to help care managers perform data-driven risk stratification or other forms of population segmentation. Such systems must help users partition patient populations based on complex sets of clinical factors, with the goal of identifying patient subgroups that would be most responsive to various forms of clinical intervention. This type of problem is often framed as a cost-benefit analysis with the goal of optimizing the return for a given policy or intervention.

Unlike point-of-care or patient-facing tools, the users of population management systems can often dedicate significant and sustained effort on a given analysis. Moreover, these users must often develop and test new hypotheses. Therefore, exploratory visual analysis tools—including those with more sophisticated visual designs—have the potential to produce significant value and impact patient outcomes.

Health Outcomes Research

Health outcomes research practitioners study even broader populations. Unlike population management professionals, who typically focus on populations within a single institution, health outcomes researchers are often focused on analyzing overall populations within a community. For example, epidemiologists in public health departments gather data across geographic regions to study outbreaks and risk factors within their target populations. Large-scale disease surveillance systems and clinical data networks have been built to help study diseases such as cancer, heart failure, and diabetes at the scale of many millions of patients. This approach is often described as "secondary use," reflecting the analysis of data that is collected primarily for uses such as care delivery or billing. Similarly, pharmaceutical companies are working to monitor after-market data about medications to track side effects, off-label uses, and more, using what the drug industry often calls "real-world evidence."

These use cases are similar to population management, except that researchers are typically tasked with deriving discoveries and insights that generalize across broad populations. Such users, therefore, can greatly benefit from exploratory visual interfaces, but they must have tools that help them assess a range of data-quality issues that emerge as analyses span data from multiple, distinct data sources collected from sites with patient populations that are highly heterogeneous and not necessarily representative of the population at large.

Data Complexity

In several of the scenarios outlined in the last section, users endeavor to draw insights from datasets representing vast numbers of patients. In other use cases, the focus is on data for a single patient, although it is often contextualized with data obtained from sets of other similar patients. As the patient datasets that feed these tools grow in size with each passing day, visualization tools are increasingly faced with overcoming many of the same big data challenges found in other fields such as Internet advertising, security, and military intelligence.

However, while the raw volume of data can be challenging, it is perhaps the easiest to address with existing technologies. First, visualization is inherently well-suited to summarize large volumes of data. Many common visual representations, such as bar charts and other proportional symbols, work well regardless of the number of data entities being visualized. Second, many of the technologies developed to help manage ever-growing data stores, such as Hadoop, Spark, or BlinkDB, can be adopted in the healthcare field. Finally, recent efforts at developing progressive approaches to visual analytics,[10] which attempt to balance the timeliness of results with the realities of long-running computational analytics, can play a key role (see Figure 2).

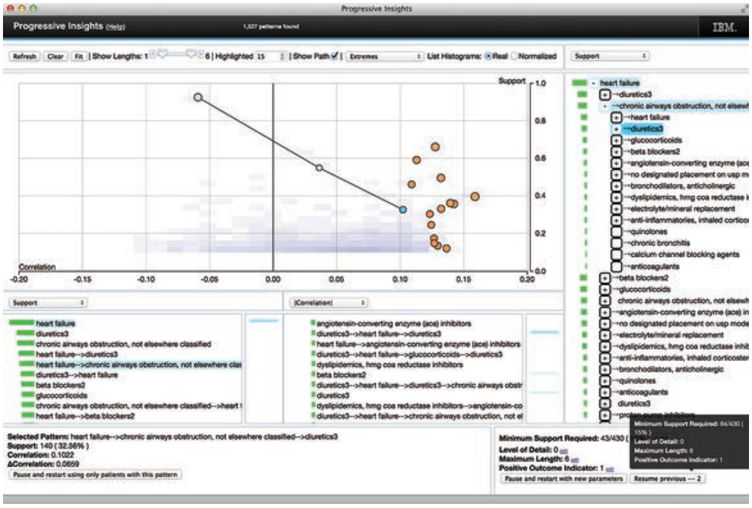


Figure 2. Progressive visual analytics approaches, such as Progressive Insights[10] shown here, can help maintain interactive update rates for data visualization in the context of long-running computational analytics. This is common when analyzing large-scale health datasets.

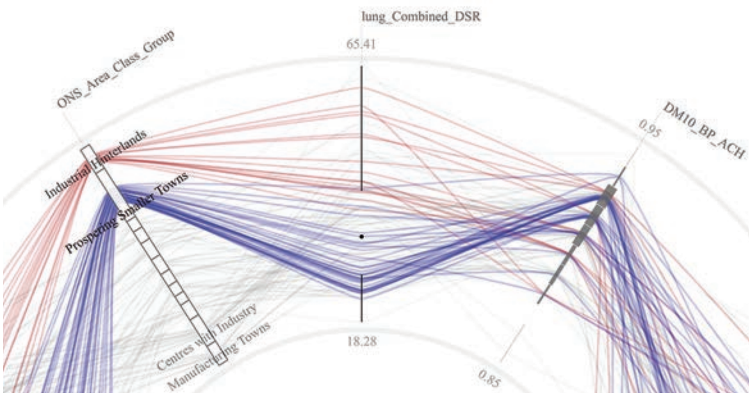


Figure 3. Data-type-dependent axis representations in a multivariate visualization of data from the United Kingdom's National Health Service.[11] Axes represent categorical, continuous, and discrete-valued data.

Data Variety

Perhaps the greatest challenge for visualization in the healthcare domain is the variety of data available. EHR systems include data such as demographics, diagnoses, procedures, medications, lab values, unstructured clinical notes, radiology results, genomic data, and more. Each of these types of data by itself contains vast numbers of variables. For example, there are approximately 68,000 unique diagnosis codes in the ICD-10-CM coding system (the current standard for diagnosis codes within the US). Across all data types, the number of variables can grow to the hundreds of thousands. Visualizations must therefore be designed to enable users to navigate this complex variable space despite that fact that the number of visualized dimensions can never approach the true dimensionality of the dataset.

In addition to the number of dimensions, the variety of data representations within those dimensions is vast. Visualizations must be able to effectively handle and display multiple data types (such as numeric, categorical, and hierarchical) and then show relationships between them[11] (see Figure 3). Effective summarization and prioritization techniques must also be adopted, including hierarchical and temporal aggregation to support levels of detail in a user's analysis[12] (see Figure 4).

Finally, many of the variables are temporal in nature, with medical events unfolding over time as patients' conditions evolve. For many use cases, effective visualizations must therefore be able to reveal interesting temporal patterns involving discrete events, interval events, and various measures recorded repeatedly over time[13] (Figure 5).

Figure 4. Flow-based visualization methods have been used in several different projects to analyze temporal event data, including the Outflow visualization shown here.[12] This example shows variations in the order of symptom onset for a cohort of heart failure patients and the medical outcomes associated with each subgroup.

Data Quality

Data quality is another critical issue with enormous consequence in the medical domain. Missing or invalid data is common, often produced by improper data entry or a lack of system interoperability. Adding to the challenge is that the absence of medical events is typically not recorded in the EHR. This makes it difficult to distinguish between missing events and events that have not occurred. In addition, institutions often optimize coded medical data for billing purposes, making it less reliable for subsequent analysis. This is especially true for claims data, which is often easy to obtain but of limited quality due to bias and lack of specificity. (Claims data is generated for billing rather than clinical care, so it is less clinically specific.) Visualizations must be designed to work effectively with these data-quality issues and should highlight deficiencies in the underlying data to users to consider as part of their analysis.

Finally, a large portion of electronic medical data is contained in unstructured fields, such as clinical notes and medical images. Prior to visualization, these resources are often analyzed with natural-language processing and image-analysis algorithms to produce structured annotations. However, these algorithms introduce their own uncertainties into the resulting data. Uncertainty visualization—a longstanding challenge facing the visualization community—is another critical issue that must be addressed.

Data Heterogeneity

Heterogeneity of data sources is another growing challenge in the medical domain, especially in countries with decentralized information systems. For example, EHR systems in the US are typically maintained independently by each provider organization, with limited interoperability or joint governance across institutions. Many efforts are underway to federate data across these disjointed systems to enable epidemiological surveillance and population health research activities. However, bridging the political and technological gaps between different data silos can be daunting. Additionally, federated collections of data collected under different standards and practices contain higher levels of heterogeneity within the data. This makes the already complex challenge of data variety even more challenging.

In addition, there is the challenge of linking data across different sources to construct a combined longitudinal record for individual patients. Data from different sources will often be in different formats, necessitating flexibility in the data models used for integrating and visualizing this information. Moreover, a lack of standard patient identifiers in many health systems means that identity resolution must often be done probabilistically. Linking across sources in this way is difficult in many domains, but it is especially hard when dealing with personal health information, where privacy issues are of concern and can limit the amount of information available to perform the linking.

Finally, a variety of new and emerging big data sources are becoming available, which further complicates the data landscape. From genomics to social media data to new data-gathering personal medical devices, these new sources exacerbate all the data complexity challenges we've already discussed. For example, mobile health devices such as the Fitbit and Apple Watch gather longitudinal data from patients over time periods when no data would normally be available. Although these devices promise to improve health care in various ways, they also produce vast amounts of additional data for clinicians—who are already overwhelmed with data—to review at the point of care. Moreover, the data produced is often of questionable quality, with a lack of standardized collection practices (for example, users forgetting to wear a device, short battery life, and improper fit or usage).

Statistical Rigor

Across the full spectrum of use cases—from single patient point-of-care treatment decisions to population-focused epidemiological studies—the healthcare domain places unique demands on visualization tools in terms of statistical rigor. A doctor's choice about which medication to prescribe, or which diagnosis to make, can have life-or-death consequences. Similarly, conclusions drawn about medical outcomes based on data in federated collections representing millions of patients can impact the well-being of vast numbers of people. Put simply, the cost of a wrong conclusion when prescribing a medication is potentially much higher than mistakes in targeting an advertisement.

As a result, visualizations that help users identify "interesting" trends or insights, while useful in some contexts, are often insufficient. At the same time, purely statistical approaches have their own limitations, including a lack of interpretability and contextualization. For tools to be adopted in practice, they must combine best practices from both the statistical and visualization disciplines. Moreover, they must satisfy the high standards for validity that the healthcare domain demands. In particular, visualization tools that provide actionable insights must address the same issues of validity that determine the design of traditional clinical trials.[14]

It can be tempting to assume that the trend toward big data can address this challenge. By capturing real-world data with rich detail and at a large scale, it is hoped that sufficient data can be obtained to yield data-driven insights that can reliably inform decision making. However, as the well-publicized inaccuracies with Google Flu Trends have shown, the large volume of data available only directly addresses the issue of sample size.

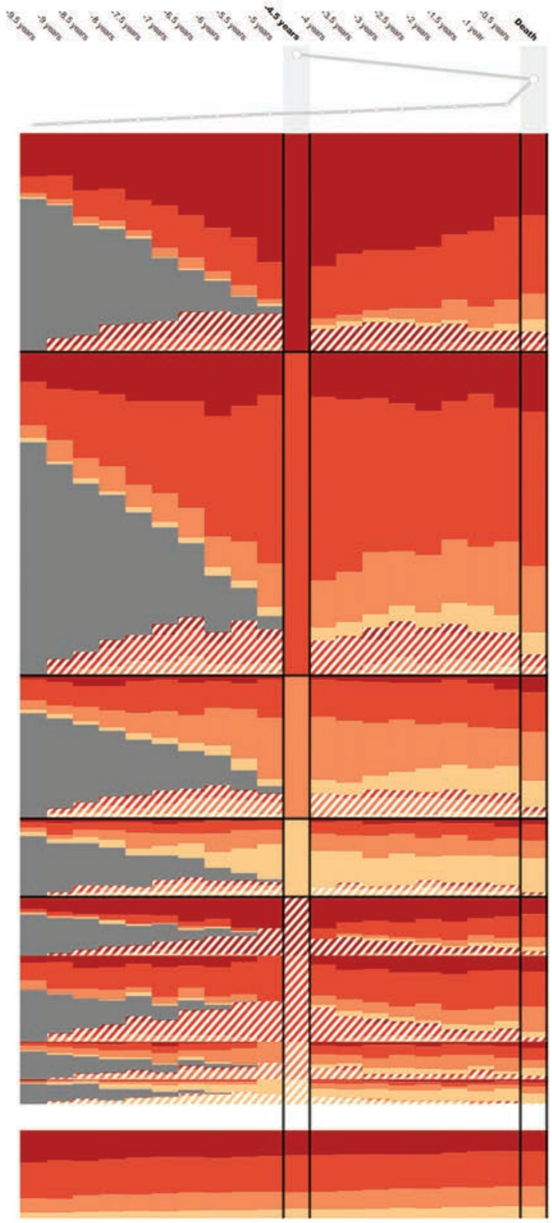


Figure 5. Repeated measures over time. This path map temporal visualization of hemoglobin A1c levels in thousands of diabetic patients incorporates data aggregation as well as missing data display via a striped pattern.[13]

Spurious Relationships

A common problem when dealing with big data is spurious relationships, when two events or variables incorrectly appear to have a causal relationship. In fact, when enough data is present, it is not only likely, but expected, that some variables will correlate closely with others based only on chance. In these cases, it is easy for users to improperly infer that a meaningful relationship exists.

For this reason, visualization systems should not rely solely on statistical significance or visually apparent trends. Instead, visualizations that present relationships between data entities and variables should provide effective tools that—based on appropriate statistical mechanisms that assess repeatability and generalizability—help users identify and correctly discount relationships most likely to be spurious.

Selection Bias

Selection bias also abounds when dealing with large amounts of medical data. This issue manifests itself in two different problems. First, differences between a general population and the sample population loaded into a visualization system can result in nongeneralizable insights. For example, the profile for a population of patients treated by a private hospital may not be the same as one treated in a public hospital. Similarly, patients treated at a specific type of clinic may not reflect the general population. This form of selection bias means that visualizations based on a specific dataset from a nonrepresentative source may not produce generalizable results. Datasets must therefore be compared with baseline population samples to determine the generalizability of any visualized results.

Second, many interactive visualization tools are designed specifically to enable users to filter data "on demand" as part of an exploratory, ad hoc cohort selection process. This is a key value proposition offered by many visualization systems. Although selection bias is already a problem for activities such as selecting a cohort for clinical trials, the rapid ad hoc filtering available in many interactive visualizations introduces the risk of selection bias at each step. Visualization tools should provide mechanisms for helping users assess this form of selection bias and, ideally, produce more informed and representative data selections. Moreover, this second form of bias further motivates the need to use baseline population representations to contextualize exploratory analysis or data selection.

Interactive data visualization in the healthcare domain presents many exciting opportunities, along with many challenges. These challenges provide an ideal playground for visualization researchers to advance the field. Moreover, with increased attention to this domain, and through close collaboration with healthcare practitioners, researchers have the opportunity to greatly impact the state of the art in healthcare, enabling better treatment at reduced costs, resulting in improved patient outcomes, and potentially saving lives.

References

1. D. Charles, M. Gabriel, and T. Searcy, "Adoption of Electronic Health Record Systems among U.S. Non-Federal Acute Care Hospitals: 2008–2014," ONC data brief, Office Nat'l Coordinator of Health Information Technology, 2015; https://www.healthit.gov/sites/default/files/data-brief/2014HospitalAdoptionDataBrief.pdf.
2. C. Schoen et al., "A Survey of Primary Care Doctors in Ten Countries Shows Progress in Use of Health Information Technology, Less in Other Areas," *Health Affairs*, vol. 31, no. 12, 2012, pp. 2805–2816.
3. T.B. Murdoch and A.S. Detsky, "The Inevitable Application of Big Data to Health Care," *J. Am. Medical Assoc.*, vol. 309, no. 13, 2013, pp. 1351–1352.
4. L.M. Etheredge, "A Rapid-Learning Health System," *Health Affairs*, vol. 26, no. 2, 2007, pp. w107–w118.
5. A. Rind et al., "Interactive Information Visualization to Explore and Query Electronic Health Records," *Foundation and Trends in Human-Computer Interaction*, vol. 5, no. 3, 2013, pp. 207–298.
6. B. Shneiderman, C. Plaisant, and B.W. Hesse, "Improving Healthcare with Interactive Visualization," *Computer*, vol. 46, no. 5, 2013, pp. 58–66.
7. J.J. Caban and D. Gotz, "Visual Analytics in Healthcare: Opportunities and Research Challenges," *J. Am. Medical Informatics Assoc.*, vol. 22, 2015, pp. 260–262.
8. V.L. West, D. Borland, and W.E. Hammond, "Innovative Information Visualization of Electronic Health Record Data: A Systematic Review," *J. Am. Medical Informatics Assoc.*, vol. 22, 2015, pp. 330–339.
9. B. Gallego et al., "Bringing Cohort Studies to the Bedside: Framework for a 'Green Button' to Support Clinical Decision-Making," *J. Comparative Effectiveness Research,* vol. 4, no. 3, 2015, pp. 191–197; doi:10.2217/cer.15.12.
10. C.D. Stolper, A. Perer, and D. Gotz, "Progressive Visual Analytics: User-Driven Visual Exploration of In-Progress Analytics," *IEEE Trans. Visualization and Computer Graphics*, vol. 20, no. 12, 2014, pp. 1653–1662.
11. D. Borland, V.L. West, and W.E. Hammond, "Multivariate Visualization of System-Wide National Health Service Data Using Radial Coordinates," *Proc. Workshop on Visual Analytics in Healthcare* (VAHC), 2014; www.visualanalyticshealthcare.org/docs/VAHC2014_proceedings.pdf.
12. K. Wongsuphasawat and D. Gotz, "Exploring Flow, Factors, and Outcomes of Temporal Event Sequences with the Outflow Visualization," *IEEE Trans. Visualization and Computer Graphics*, vol. 18, no. 12, 2012, pp. 2659–2668.
13. D. Borland et al., "Path Maps: Visualization of Trajectories in Large-Scale Temporal Data," *Poster Abstracts of IEEE VIS*, 2015.
14. P. Jüni, D.G. Altman, and M. Egger, "Assessing the Quality of Controlled Clinical Trials," *BMJ*, vol. 323, 2001, pp. 42–46; www.bmj.com/content/323/7303/42.

David Gotz *is the assistant director for the Carolina Health Informatics Program and an associate professor of information science in the School of Information and Library Science at the University of North Carolina at Chapel Hill. Contact him at gotz@unc.edu.*

David Borland *is a senior visualization researcher with the Renaissance Computing Institute (RENCI) at the University of North Carolina at Chapel Hill. Contact him at borland@renci.org.*

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

This article originally appeared in
IEEE Computer Graphics and Applications
vol. 36, no. 3, 2016.

Move Your Career Forward

IEEE Computer Society Membership

Explore Emerging Technology Resources

Build Your Knowledge

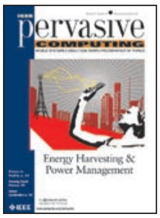
IEEE Transactions on Emerging Topics in Computing

TETC will keep you up-to-date with the most recent papers on emerging aspects of computer science, computing technology, and computing applications.

IEEE computer society ROCK STARS OF EMERGING TECHNOLOGIES

Rock Stars of Emerging Technologies
18 May 2017
Austin, TX, USA

With the host of emerging technologies now coming to market, this is one of the most exciting and potentially disruptive times to be in business. At this conference, executives from industry-leading companies will provide key insights on six key technologies: artificial intelligence, machine learning, deep learning, distributed analytics and network visualization, quantum computing, and 5G.



IEEE Pervasive Computing

Designed for researchers, practitioners, and educators, *IEEE Pervasive Computing* magazine explores the role of computing in the physical world as characterized by technological approaches such as the Internet of Things and ubiquitous computing.

IEEE Transactions on Big Data

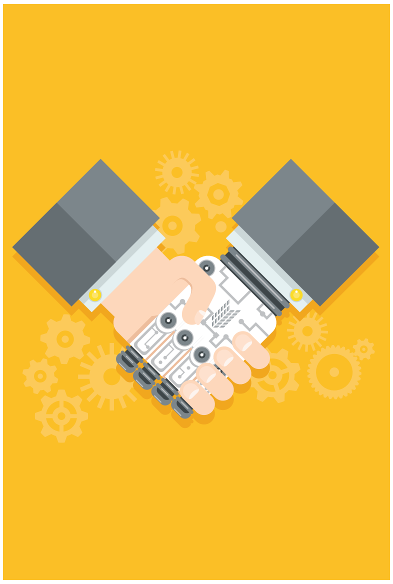
TBD's peer-reviewed articles present innovative cross-disciplinary research results and applications for big data.

FOR DIRECT LINKS TO THESE RESOURCES, VISIT

www.computer.org/edge-resources

IEEE computer society

EDITOR: George Strawn, gostrawn@gmail.com



Data-Intensive Science

George Strawn, *US National Academies of Sciences, Engineering, and Medicine*

Last issue's IT and Future Employment column was on data science. Here, I look at data science applied specifically to scientific disciplines, which is now called data-intensive science. *Data* from observation and experimentation has been the foundation of science at least since the 17th century. *Information* in the form of articles that summarize the new knowledge gained from observation and experimentation (and from theory) has been the vehicle that has made science a public endeavor.

Elizabeth Eisenstein in her book *The Printing Press as an Agent of Change*[1] argued that printed information was a necessary prerequisite for the scientific revolution to occur. Now, it can be argued that computed information is creating a second scientific revolution based on digital data and information.[2] This can be seen in the exciting developments surrounding big data, which increasingly can be processed by computers to produce new knowledge. Research data and information are a major source of big data. Especially in the case of science, big data can mean big volume, big velocity, or big variety. The so-called "long tail of science" of millions of small scientific datasets constitutes big data that at present cannot be meaningfully combined for processing.

This column will highlight some of the policy, research, and infrastructure developments that are in the process of turning data-intensive science into a significant employment opportunity, both for scientists themselves and for the IT professionals who assist them in their research endeavors. Finally, I will mention several scientific disciplines in which data-intensive science has taken hold and point to one concern that needs to be addressed.

Policy Developments

One policy goal, public access to research data and information, was recently initiated when the US federal government required that all grant recipients make both the articles and data that result from their supported research publicly available (https://www.whitehouse.gov/sites/default/files/microsites/ostp/ostp_public_access_memo_2013.pdf). Another critical policy (or practice) goal would be for the academic research community to give as much promotion and tenure credit for publishing data as for publishing articles. The publishing of data has begun,[3] but the credit for doing so is still developing. This pair of policies would provide the necessary incentives to make much research data and information publicly available. In all likelihood, much other research would then follow the lead of federally sponsored research.

Data Science Research

Additional data science research will be required to make published data "fully useful." One definition of fully useful has recently been suggested,[4] and comes with its own acronym: FAIR data. The acronym is decrypted as findable, accessible, interoperable, and reuseable. In other words, the data should be (automatically!) reusable, and that usually requires that they be findable, accessible, and interoperable. Perhaps the most challenging task here is to develop schemes (undoubtedly involving metadata) that enable the interoperability of heterogeneous data. An early example of such research is the digital object architecture[5] developed by Bob

 2469-7087/17/$33.00

Kahn and his colleagues at CNRI. Another example is the Semantic Web (https://en.m.wikipedia.org/wiki/Semantic_Web), which can produce a web of data to be read by computers, whereas the original Web produced a web of documents to be read by humans. An early example of interoperability of research article information is Semantic Medline[6] developed by Tom Rindflesch and his colleagues at the National Library of Medicine. These systems enable computers to process data and information in ways that previously could not be done without human involvement.

An interesting and necessary variation on "open" data-intensive science is that schemes need to be developed that will allow reuse of data that—for privacy, proprietary, or security reasons—cannot be made publicly accessible. An example is data that involve human subjects, which are subject to privacy restrictions. Aggregations of such data need to be constructible without revealing the individual data records. Anonymizing techniques are also a possible way to make private information publicly accessible, but concerns about deanonymizing the data are delaying use of this technique.

Infrastructure for the Second Scientific Revolution

In addition to the polices and research needed to accomplish the goals of FAIR data, there remains the infrastructure issue of where to store the data. Recent developments in cloud computing provide one ready answer to this need. In Europe, the Open Science Cloud project (https://ec.europa.eu/research/openscience/index.cfm?pg=open-science-cloud) and in the US, the Internet2 Net+ project (www.internet2.edu/vision-initiatives/initiatives/internet2-netplus/) are early examples of the application of cloud computing to the storage of research data and information.

Progress Toward Data-Intensive Science

Some disciplines, such as astronomy, are leaders in establishing metadata standards that enable reuse of their data. In fact, it is said that use of these standards has redefined the discipline (www.ivoa.net), enabling all conforming observations of an area of the sky to be digitally overlaid. Now, not only can astronomical phenomena be captured, but also systematic changes in those phenomena can be studied. Another well-known example of the development and use of fair data is the human genome project. By standardizing the data formats and requiring researchers to populate a public genomic database (and by developing increasingly automated processing techniques), the human genome was decoded, and a new era of biology was begun.

Many disciplines are developing new approaches and standards for data as we enter the era of data-intensive science. Some observers have called data-intensive science the fourth paradigm (http://research.microsoft.com/en-us/collaboration/fourthparadigm/), the first three paradigms of science being experimental/observational, theoretical, and computational. But a danger exists in that disciplines tend to develop their data standards in isolation. To be of greatest use, the reusability of data needs to be interdisciplinary, not just within a discipline. This will require that research data standards be developed with an eye toward *interdisciplinary reusability*. Perhaps a subfield of applied data science will emerge that will assist the disciplines to develop their data standards with this goal in mind. The "new" field of data science provides a unifying framework for the individual disciplines, each of which has its own data and its own established processes for dealing with that data (https://dl.dropboxusercontent.com/u/23421017/50YearsDataScience.pdf).

We currently have the vision of open science and some specific examples of using data to reinvent disciplines (such as astronomy and genetics). But the problem of the general interoperability of heterogeneous data is harder than the problem of interoperability of heterogeneous networks (which was solved by the Internet) and remains to be solved. Nevertheless, more disciplines are redefining themselves through data-intensive science. The research to increase the scope of what's possible is active, the infrastructure to store the data is under development, and the policies to incentivize scientists to publish their data are emerging. A new scientific revolution is in the offing.

References

1. E.L. Eisenstein, *The Printing Press as an Agent of Change*, Cambridge Univ. Press, 1980.
2. D.S. Robertson, *The New Renaissance: Computers and the Next Level of Civilization*, Oxford Univ. Press, 1998.
3. L. Candela et al., "Data Journals: A Survey," *J. Assoc. Information Science and Technology*, vol. 66, no. 9, 2015, pp. 1747–1762.
4. M.D. Wilkinson et al., "The FAIR Guiding Principles for Scientific Data Management and Stewardship," *Scientific Data*, vol. 3., 2016, article no. 160018; www.nature.com/articles/sdata201618.
5. "Lost Something on the Internet? Never Again with New Digital Object (DO) Architecture," ITU blog, 6 Jan. 2014; https://itu4u.wordpress.com/2014/01/06/lost

-something-on-the-internet-never-again-with-new-digital-object-do-architecture/.

6. T.C. Rindflesch et al., "MEDLINE: An Advanced Information Management Application for Biomedicine," *Information Services and Use*, vol. 31, nos. 1–2, 2011, pp. 15–21.

***George Strawn** is the director of the Board for Research Data and Information at the US National Academies of Sciences, Engineering, and Medicine. He is the former director of the National Coordination Office for the Networking and Information Technology Research and Development Program (NITRD). Contact him at gostrawn@gmail.com.*

This article originally appeared in IT Professional *vol. 18, no. 5, 2016.*

CALL FOR NOMINEES

Education Awards Nominations

Taylor L. Booth Education Award

A bronze medal and US$5,000 honorarium are awarded for an outstanding record in computer science and engineering education. The individual must meet two or more of the following criteria in the computer science and engineering field:

- Achieving recognition as a teacher of renown.
- Writing an influential text.
- Leading, inspiring or providing significant education content during the creation of a curriculum in the field.
- Inspiring others to a career in computer science and engineering education.

Two endorsements are required for an award nomination.

See the award information at:
www.computer.org/web/awards/booth

Computer Science and Engineering Undergraduate Teaching Award

A plaque, certificate and a stipend of US$2,000 is awarded to recognize outstanding contributions to undergraduate education through both teaching and service and for helping to maintain interest, increase the visibility of the society, and making a statement about the importance with which we view undergraduate education.

The award nomination requires a **minimum of three endorsements**.

See the award details at:
www.computer.org/web/awards/cse-undergrad-teaching

Deadline: 15 October 2017
Nomination Site: awards.computer.org

IEEE

IEEE computer society

Careers in Emerging Technologies: Databases and Data Science

For this issue of *ComputingEdge*, we asked Andy Pavlo—assistant professor of databaseology in Carnegie Mellon University's Computer Science Department—about career opportunities in emerging technology fields involving databases and data science. Pavlo's research interests are database management systems—specifically main memory, nonrelational, and transaction-processing systems—and large-scale data analytics. He authored the article "Emerging Hardware Trends in Large-Scale Transaction Processing" in *IEEE Internet Computing*'s May/June 2015 issue.

ComputingEdge: What careers in emerging technologies in your field will see the most growth in the next several years?

Pavlo: Artificial intelligence, more specifically machine learning, will continue to be the hot growth area for the foreseeable future in database- and data-science-related fields. Developers who can design high-performance systems to support complex, data-intensive applications will surely be in demand for at least several years.

ComputingEdge: What would you tell college students to give them an advantage over the competition?

Pavlo: No company or organization starts a new software project from scratch. Thus, it's good to have the ability to work on existing code bases with little or no guidance or documentation. The most ideal employees are those who can start quickly on a project that consists of a large amount of existing code they didn't write. The good way to learn this skill is through practice.

ComputingEdge: What advice would you give people changing careers midstream?

Pavlo: You must always work hard. And you have to stay up-to-date with the latest database systems, machine-learning tools, and data-analysis frameworks. Luckily, we live in an era where everyone is releasing their software as open source, so it's easier for people to try things out at home. The best way to pick up new skills is to pick a hobby project and then build it out using a new piece of software that you want to learn more about.

 Published by the IEEE Computer Society 2469-7087/17/$33.00

ComputingEdge: What do you consider to be the best strategies for professional networking?

Pavlo: You need to be visible. Simply making a LinkedIn page isn't enough. You must advertise what you have to offer. This means you should write a blog, build out your GitHub portfolio, contribute to open source projects, attend and give talks at meet-ups, and/or volunteer for hackathons. All of this shows potential employers that you are enthusiastic about computers and technology. Every little bit helps.

ComputingEdge: What should applicants keep in mind when applying for emerging-technology jobs?

Pavlo: The field is moving fast, but having a good computer-science foundation will serve you well no matter what the current technology trend is.

ComputingEdge's Lori Cameron interviewed Pavlo for this article. Contact her at l.cameron@computer.org if you would like to contribute to a future ComputingEdge article on computing careers. Contact Pavlo at pavlo@cs.cmu.edu.

CONFIGURATION SPECIALIST. General Motors, Detroit, MI. Improve connected customer exp. Desg, style, and dvlp Chrome extensions, using AngularJS, BootStrap and LESS, to validate search algorithm changes and configuration changes in MultiChannel Engine incldg browser responsive desg. Solve and manage dependencies using Bower and create automated builds and testing using Grunt and Jasmine. Build interface integrating merchant API with internal apps to harness Point of Interest information for pre-matching and serving offers to consumers using Python 3.5, Pandas and NumPy libraries. Master, Computer Science or Electrical Engrg. 12 mos exp as Computer Programmer or related, designing, styling, and dvlpg Chrome extensions, using AngularJS, BootStrap and LESS, to validate search algorithm changes and configuration changes in MCE incldg browser responsive desg. Mail resume to Alicia Scott-Wears, GM Global Mobility, 300 Renaissance Center, MC:482-C32- D44, Detroit, MI 48265, Ref#35988.

TENURE-TRACK/TENURED FACULTY POSITIONS. The Department of Computer Science at Purdue University is in a phase of significant growth. Applications are solicited for seven tenure-track and tenured positions at the Assistant, Associate and Full Professor levels. Outstanding candidates in all areas of computer science will be considered. Review of applications and candidate interviews will begin in October 2016, and will continue until the positions are filled. The Department of Computer Science offers a stimulating academic environment with research programs in most areas of computer science. Information about the department and a description of open positions are available at http://www.cs.purdue.edu. Applicants should hold a PhD in Computer Science or a related discipline, have demonstrated excellence in research, and strong commitment to teaching. Successful candidates will be expected to conduct research in their fields of expertise, teach courses in computer science, and participate in other department and university activities. Salary and benefits are competitive, and Purdue is a dual-career friendly employer. Applicants are strongly encouraged to apply online at https://hiring.science.purdue.edu. Alternatively, hardcopy applications can be sent to: Faculty Search Chair, Department of Computer Science, 305 N. University Street, Purdue University, West Lafayette, IN 47907. A background check will be required for employment in this position. Purdue University is an EEO/AA employer. All individuals, including minorities, women, individuals with disabilities, and veterans are encouraged to apply.

CLOUDERA, INC. is recruiting for our Palo Alto, CA office: Solutions Architect: drive technical customer conversations, understand customer reqs & transfer acquired knowledge into a technical plan of action. Travel required. Telecommuting allowed. Mail resume w/job code #37212 to: Cloudera, Attn.: HR, 1001 Page Mill Rd., Bldg. 2, Palo Alto, CA 94304.

TECHNOLOGY

Linkedin Corp.

has openings in our

Sunnyvale, CA location for

Talent Integration Solution Developer

(6597.1460)

Design, develop, test, deploy, support, & enhance custom web portals & integration solutions and seamlessly connect LinkedIn enterprise systems.

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

INFO TECH

Oracle America, Inc.

has openings for

IT DIRECTOR

positions in **Atlanta, GA.**

Job duties include: Drive innovation and manage technical projects. Regularly interact with senior management on matters concerning several functional areas, and/or customers. Provide technical guidance, career development, performance assessment, and mentoring to team members. May telecommute from home.

Apply by e-mailing resume to raj.raina@oracle.com, referencing 385.13939.

Oracle supports workforce diversity.

TECHNICAL

Oracle America, Inc.

has openings for

TECHNICAL ANALYST

positions in **Lehi, UT**.

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

Apply by e-mailing resume to wael.ibrahim@oracle.com, referencing 385.19776.

Oracle supports workforce diversity.

 2469-7087/17/$33.00 © 2017 IEEE

CLOUDERA, INC. is recruiting for our San Francisco, CA office: Sr. Solutions Consultant: work with Cloudera customers to install & implement CDH platform & develop solutions on the Cloudera infrastructure. Travel required. Mail resume w/job code #34068 to: Cloudera, Attn.: HR, 1001 Page Mill Rd., Bldg. 2, Palo Alto, CA 94304.

PROGRAMMER ANALYST. IT company in Matawan NJ seeks Programmer Analysts responsible for interacting with clients to gather requirements, preparing technical and functional specifications, creating various types of tests, troubleshooting, maintaining, fine tuning applications, analyzing, designing, implementing web based client/server applications utilizing knowledge of Java, J2EE, Servlets, JSP, Spring, Hibernate, JDBC, Java Script, XML, HTML, CSS, Oracle, SQL, WebSphere, WebLogic, Unix, Linux, Windows. Apply with 2 copies of resume to HR, Questsoft Solutions Inc, 432 State Rt.34, Ste#3A, Matawan, NJ-07747

TECHNOLOGY

Expedia, Inc.

has openings for the following positions in **San Francisco, California (various/levels/types)**:

Software Engineers (Job ID#: 728.SWE-ESF-DEC): Design, implement, and debug software for computers including algorithms and data structures.

Technical Product Managers (Job ID#: 728.TPM-ESF-DEC): Gather detailed business requirements from stakeholders and work closely with technology staff to translate requirements into functional designs and specifications.

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

FIU FLORIDA INTERNATIONAL UNIVERSITY

Florida International University is classified by Carnegie as a R1: Doctoral Universities - Highest Research Activity and recognized as a Carnegie engaged university. It is a public research university with colleges and schools that offers 196 bachelor's, master's and doctoral programs in fields such as engineering, computer science, international relations, architecture, law and medicine. As one of South Florida's anchor institutions, FIU contributes almost $9 billion each year to the local economy. FIU is Worlds Ahead in finding solutions to the most challenging problems of our time. FIU emphasizes research as a major component of its mission. FIU has awarded more than 220,000 degrees and enrolls more than 54,000 students in two campuses and three centers including FIU Downtown on Brickell, FIU@I-75, and the Miami Beach Urban Studios. FIU's Medina Aquarius Program houses the Aquarius Reef Base, a unique underwater research facility in the Florida Keys. FIU also supports artistic and cultural engagement through its three museums: Patricia & Phillip Frost Art Museum, the Wolfsonian-FIU, and the Jewish Museum of Florida-FIU. FIU is a member of Conference USA and has more than 400 student-athletes participating in 18 sports. For more information about FIU, visit http://www.fiu.edu/.

FIU's School of Computing and Information Sciences (SCIS) is a rapidly growing program of excellence at Florida International University (FIU). The School has 29 tenure-track faculty members and over 2,000 students, including over 90 Ph.D. students. The School is engaged in on-going and exciting new and expanding programs for research, education and outreach. The School offers B.S., M.S., and Ph.D. degrees in Computer Science, and M.S. degrees in Telecommunications and Networking, Cyber-security, and Information Technology as well as B.S./B.A degrees in Information Technology. NSF ranks FIU 43rd nationwide in externally-funded research expenditures. SCIS has six research centers/clusters with first-class computing and support infrastructure, and enjoys broad and dynamic industry and international partnerships.

We invite applications from exceptionally qualified faculty at all levels with particular emphasis on networking, cyber-security, computer systems or data sciences, and other related areas. Ideal candidates for junior positions should have a record of exceptional research in their early careers and a demonstrated ability to pursue and lead a research program. Candidates for senior positions must have an active and sustainable record of excellence in funded research, publications and professional service as well as demonstrated leadership in collaborative or interdisciplinary research. In addition to developing or expanding a high-quality research program, all successful applicants must be committed to excellence in teaching at both the graduate and undergraduate levels. Applications are encouraged from candidates with highly transformative research programs and seminal ideas that extend the frontiers of computing and networking across other disciplines. A Ph.D. in Computer Science or related disciplines is required.

HOW TO APPLY:

Qualified candidates are encouraged to apply to Job Opening ID **(Job Opening ID # 512441)** at ***facultycareers.fiu.edu*** and attach a cover letter, curriculum vitae, statement of teaching philosophy, research statement, etc as *individual attachments*. Candidates will be requested to provide names and contact information for at least three references who will be contacted *as determined by the search committee*. To receive full consideration, applications and required materials should be received by December 31, 2016. Review will continue until position is filled.

FIU is a member of the State University System of Florida and an Equal Opportunity, Equal Access Affirmative Action Employer. All qualified applicants will receive consideration for employment without regard to race, color, religion, sex, national origin, disability status, protected veteran status, or any other characteristic protected by law.

VIMO (DBA GETINSURED.COM) seeks User Interface Dvlper. to work w/ Engin./ Dsgn. team to trnslte. visual dsgns. to prdction. ready screens in HTML/CSS/ JS. Build new/improve interaction patterns. Participate in dsgn./troublesht. SW. Resumes to Leslie Sullivan, GetInsured.com, 1305 Terra Bella Ave., Mountain View, CA 94043.

SENIOR SOFTWARE ENGINEER. Warren, MI, General Motors. Desg, dvlp &test embedded soft &device driver for Android based psgr vehicle infotainment sys on TI Jacinto 6 ARM processor platform using embedded C/C++/Java language, CAN bus analyzer tools, Android framework, Linux operating sys, MicroC/ OS-II RTOS &QNX real time operating sys. Desg &dvlp AUTOSAR based soft using Vector Inc tools DaVinci Configurator Pro for configuring BSW Module &DaVinci dvlpr tools to model soft cmpt for GM Global B architecture. Master, Electronics Engrg or Computer Science. 3 years' experience as Technical Architect or Engineer. Will accept bachelor or foreign equiv degree in Electronics Engrg or Computer Science, followed by at least 5 yrs of progressive exp in the specialty, in lieu of the required education &exp. Will accept any equally suitable combination of education, training, &/or exp which would qualify applicant to perform job offered. Required exp must include 36 mos exp desging, dvlpg &testing embedded soft &device driver for psgr vehicle infotainment sys on ARM processor platform using embedded C/C++ &/or Java language, CAN bus analyzer tools &RTOS. Mail resume to Alicia Scott-Wears, GM Global Mobility, 300 Renaissance Center, MC:482-C32-D44, Detroit, MI 48265, Ref#2661.

IT FIRM has multiple openings in Indianapolis & client sites. **Sr. ETL Developers (#302)** - Participate in all phases of ETL dvlp lifecycle. Analyze reqmnts, translate specs. Dsgn, dev, test ETL framework & processes for Data Mart & Data warehouses. Use techs such as IBM Datastage, Ab Initio, Oracle, Unix, Teradata, DB2. Reqs Bachelor's w/5 yrs progr exp or Master's w/3 yrs exp. Major: CS, CE or rel field. **Cognos BI Developers (#303**) - Dsng, dvlp, implmnt Cognos TM1 9.5.2 applns using Oracle, SQL, PLSQL. Gather & analyze Data Warehouse & BI reqmnts. Dsng & dvlp dynamic cubes, framework manager. Test & optimize ETL. Reqs Master's in CS, CE, Bus Admin or rel field w/2 yrs exp. Degree may be equiv or foreign equiv. Comb of edu/train'g/exp accepted. Some positions may require reloc to unanticipated client sites. Apply w/code# to HR, Brite Systems, 101 West Ohio St, #1010, Indianapolis, IN 46204. EOE.

CLOUDERA, INC. is recruiting for our Palo Alto, CA office: Design & implement large distributed systems that scale well – to petabytes of data and 10s of 1000s of nodes. Mail resume w/job code #37128 to: Cloudera, Attn.: HR, 1001 Page Mill Rd., Bldg. 2, Palo Alto, CA 94304.

MPHASIS CORP. has multi openings at various levels for the follow'g positions at its office in NY, NY & unanticipated client sites thr/o the US 1. Info. Sys. Anyst* - Ana. & provide sys req & spec. 2. SW Dvlper* - Design, dvlp & modify SW sys. 3. Sys. Architect Dvlper* - Dvlp IT architecture 4. Graphic UI Desgr* - Design UI & perform UAT 5. N/W Infra Eng* - Maintain & TRBL n/w, design, dvlp, install n/w infra appl. 6. Business Operation Anyst* - Ana bus process thru app of s/w sol. 7. IT Mgr* - Plan & manage the delivery of IT proj. 8. Enterprise Svc Engagem't Mgr* - E2E sale of IT svc/prod. 9. Eng Engagem't Mgr* - Manage & direct business integration of proj activities. 10. Mkt Dvlpt Mgr* - Promote IT svc/ prod. & impl bus plans. Must have a Bachelor/equiv and prior rel. exp, Master/equiv, or Master/equiv and prior rel. exp. Edu/exp req vary depending on position level/type. *Lead positions in this occupation must have Master/equiv+2yr or Bach/equiv+5yr progressive exp. Travel/relo req. Send resume & applied position to: recruitmentus@mphasis.com or 460 Park Ave. S., Ste# 1101, New York, NY 10016 Attn: Recruit.

SENIOR SOFTWARE ENGINEER (Multiple Positions Available) Full lifecycle application development (lifecycle of software or online app0lication development, from initial conception of idea through design, coding, testing/quality control, debugging & upkeep using various software languages such as C# ASP, NEAT, C++, Silverlight). OOA & OOD, etc. Bachelor's Degree in Computer Engineering or Computer Information System or Computer Science plus 5 yrs exp. 40 hrs/wk. Job Site & Intvu: Marina Del Rey, CA. Send resume to Solid Commerce att: A. Berkovich at alon.b@solidcommerce.com.

TECHNOLOGY

Oracle America, Inc.

has openings for

DATABASE ADMINISTRATOR

positions in **San Bruno, CA**.

Job duties include: Provide enterprise-wide, database administration support for production systems and provide DBA services to application development teams, including database design, database generation, coding, and/or database production support.

Apply by e-mailing resume to yogesh.kaudan@oracle.com, referencing 385.19404.

Oracle supports workforce diversity.

SOFTWARE

Oracle America, Inc.

has openings for

SOFTWARE DEVELOPER

positions in **Seattle, WA**.

Job duties include: Design, develop, troubleshoot and/or test/QA software.

Apply by e-mailing resume to amit.fnu@oracle.com, referencing 385.19686.

Oracle supports workforce diversity.

LECTURER IN COMPUTING AND METHEMATICAL SCIENCES, SEPTEMBER 2017. The Department of Computing and Mathematical Sciences (CMS) at the California Institute of Technology invites applications for the position of Lecturer in Computing and Mathematical Sciences. This is a (non-tenure-track) career teaching position, with full-time teaching responsibilities. The start date for the position is September 1, 2017 and the initial term of appointment can be up to three years. The lecturer will teach introductory computer science courses including data structures, algorithms and software engineering, and will work closely with the CMS faculty on instructional matters. The ability to teach intermediate-level undergraduate courses in areas such as software engineering, computing systems or compilers is desired. The lecturer may also assist in other aspects of the undergraduate program, including curriculum development, academic advising, and monitoring research projects. The lecturer must have a track record of excellence in teaching computer science to undergraduates. In addition, the lecturer will have opportunities to participate in research projects in the department. An advanced degree in Computer Science or related field is desired but not required. Applicants are encouraged to have all materials submitted by January 15, 2017, though applications will continue to be accepted until the position is filled. Please view the application instructions and apply on-line at https://applications.caltech.edu/job/cmslect The California Institute of Technology is an Equal Opportunity/Affirmative Action Employer. Women, minorities, veterans, and disabled persons are encouraged to apply.

PROCESS ENGINEER. Tupperware U.S., Inc. currently has an opening in Hemingway, SC for a Process Engineer to develop tests and implement process improvements within the manufacturing environment including revising productivity standards and designing production floor layouts. Mail resume to DeJuan Hinson, Sr. Manager, Human Resources at 248 Tupperware Road, Hemingway, South Carolina 29554, referencing Job ID 11058.18

CLEMSON UNIVERSITY

Holcombe Department of Electrical and Computer Engineering

Faculty Search in Computer Engineering

The Holcombe Department of Electrical and Computer Engineering at Clemson University is seeking applicants for multiple computer engineering tenure-track or tenured faculty positions at the rank of assistant professor or associate professor. The Department has a particular interest in applicants in the following technical areas: (1) high-performance computing with an emphasis on Big Data, high-performance networking, or accelerated computing architectures; (2) cyber security and cyber-physical system security; and (3) biomedical systems, mobile health systems, and biologically inspired cyber-physical systems. Outstanding assistant-professor candidates will be considered for the Warren Owens Assistant Professorship.

The Holcombe Department of Electrical and Computer Engineering is one of the largest and most active at Clemson, with 36 faculty members, 550 undergraduates and 200 graduate students. Many members of the faculty are known internationally; they include eight IEEE Fellows, three endowed chairs, and four named professors. Externally funded research expenditures exceeded $8.4 million in 2016. The Department and Clemson have highly successful computing-focused research programs in high-performance computing and networking; privacy, communications security, and secure control systems; and mobile health devices.

Clemson University is the land-grant institution for the State of South Carolina enrolling approximately 17,100 undergraduates and 4,300 graduate students. Five interdisciplinary colleges house strong programs in architecture, engineering, science, agriculture, business, social sciences, arts and education. A faculty of 1,400 and staff of 3,500 support 84 undergraduate degree offerings, 73 master's degree programs and 40 Ph.D. programs. An annual operating budget of approximately $956 million and an endowment of $623 million fund programs and operations. The University has externally funded research expenditures of $100 million per year. Research and economic development activities are enhanced by public-private partnerships at 3 innovation campuses and 6 research and education centers located throughout South Carolina. Clemson University is ranked 23rd among national public universities by U.S. News & World Report.

Applicants must have an earned doctorate in electrical engineering, computer engineering, or a closely related field. Applicants should submit a current curriculum vitae, statements of research and teaching strategy, and a minimum of five references with full contact information. Application material should be submitted electronically at the following Web link: https://apply.interfolio.com/39731

To ensure full consideration, applicants must apply by February 15, 2017; however, the search will remain open until the position is filled.

Clemson University is an AA/EEO employer and does not discriminate against any person or group on the basis of age, color, disability, gender, pregnancy, national origin, race, religion, sexual orientation, veteran status or genetic information. Clemson University is building a culturally diverse faculty committed to working in a multicultural environment and encourages applications from minorities and women.

locations throughout the United States. **IT Project Manager (Ref.# RTP105):** Responsible for defining IT project requirements, plans, schedules, and resources for small to large/complex projects.

RICHARDSON, TX: Customer Support Engineer (Ref.# RIC1): Responsible for providing technical support regarding the company's proprietary systems and software. **Network Consulting Engineer (Ref.# RIC20):** Responsible for the support and delivery of Advanced Services to company's major accounts. Telecommuting permitted and travel may be required to various unanticipated locations throughout the United States. **Technical Lead/Leader (Ref.# RIC55):** Lead engineering groups on projects to design, develop or test hardware or software products. **Network Consulting Engineer (Ref.# RIC898):** Responsible for the support and delivery of Advanced Services to company's major accounts. Telecommuting permitted.

ROSEMONT, IL: Consulting Systems Engineer (Ref.# ROSE17): Provide specific end-to-end solutions and architecture consulting, technical and sales support for major account opportunities at the theater, area, or operation level. Travel may be required to various unanticipated locations throughout the United States.

SAN JOSE/MILPITAS/SANTA CLARA, CA: Consulting Systems Engineer (Ref.# SJ2): Provide specific end-to-end solutions and architecture consulting, technical and sales support for major account opportunities at the theater, area, or operation level. **Principal Engineer (Ref.# SJ673):** Lead the team for developing and enhancing routing and switching protocols and features for company's products. **Customer Support Engineer (Ref.# SJ3):** Responsible for providing technical support regarding the company's proprietary systems and software. **Solutions Architect (Ref.# SJ27):** Responsible for IT advisory and technical consulting services development and delivery. **Database Administrator (Ref.# SJ43):** Design and develop diagnostic software for verification and validation in engineering and manufacturing. **Hardware Engineer (Ref.# SJ5):** Responsible for the specification, design, development, test, enhancement, and sustaining of networking hardware. **Virtual Systems Engineer (Ref.# SJ544):** Engage customers virtually via collaborative tools and technology including web and video conferencing.

PLEASE MAIL RESUMES WITH REFERENCE NUMBER TO CISCO SYSTEMS, INC., ATTN: V51B, 170 W. Tasman Drive, Mail Stop: SJC 5/1/4, San Jose, CA 95134. No phone calls please. Must be legally authorized to work in the U.S. without sponsorship. EOE.

www.cisco.com

David J. DeWitt Chair in Computer Sciences

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

University of Wisconsin-Madison

The Department of Computer Sciences is pleased to announce the establishment of the David J. DeWitt Chair in Computer Sciences, endowed by the generosity of alumnus Dr. Rakesh Agrawal. Applications and nominations are currently being accepted.

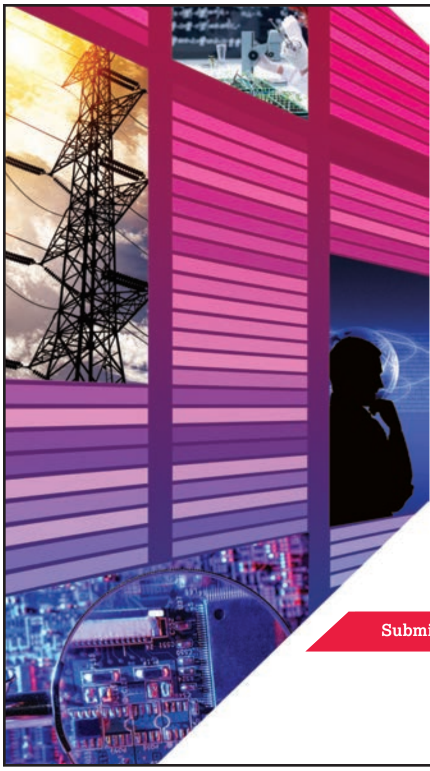
Scholars with an established track record in the area of database systems are encouraged to apply. Appointment will be as full professor with tenure. While we are especially interested in candidates with a strong "systems" focus, any scholar whose primary publication venues have been ACM SIGMOD, VLDB, ACM PODS and ACM SIGKDD is encouraged to apply.

The DeWitt Chair shall be an internationally renowned researcher and educator committed to benefitting society through scientific breakthroughs, technological innovation and the democratization of knowledge acquisition and diffusion. The recipient will be a key member of UW-Madison's database research group and join a highly ranked CS department with over $20M of research activity per year, located in a vibrant, livable city that Forbes has identified as one of the fastest-growing technology hubs in the U.S.

Additional information can be viewed at
http://www.cs.wisc.edu/about/employment

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

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



CALL FOR STANDARDS AWARD NOMINATIONS

IEEE COMPUTER SOCIETY HANS KARLSSON STANDARDS AWARD

A **plaque** and **$2,000 honorarium** is presented **in recognition of outstanding skills and dedication to diplomacy, team facilitation, and joint achievement in the development or promotion of standards** in the computer industry where individual aspirations, corporate competition, and organizational rivalry could otherwise be counter to the benefit of society.

NOMINATE A COLLEAGUE FOR THIS AWARD!

DUE: 15 OCTOBER 2017

- Requires 3 endorsements.
- Self-nominations are not accepted.
- Do not need IEEE or IEEE Computer Society membership to apply.

Submit your nomination electronically: awards.computer.org | Questions: awards@computer.org

IEEE   IEEE computer society

TECHNOLOGY

Help build the next generation of systems behind Facebook's products.

Facebook, Inc.

currently has the following openings:

Openings in **Menlo Park, CA (multiple openings/various levels)**:

Application Engineer (8510J) Develop and maintain integrated, scalable, corporate applications. Manage the employee identity management system supporting employees across Facebook. **Product Support Specialist (7673J)** Monitor the technical quality and health of Facebook products and provide clear direction on top priority issues. **Security Engineer Manager (2806J)** Lead a team of security engineers and data scientists to research and understand the latest criminal safety trends and identify and consult on the design of infrastructure to detect and investigate abuse. **Engineering Manager (4188J)** Drive engineering effort, communicate cross-functionality, and be a subject matter expert; and/or perform technical engineering duties and oversee a team of engineers. **Front End Engineer (8437J)** Work with Product Designers to implement the next generation of Company's products. Build efficient and reusable front-end abstractions and systems. **UX Research Manager (3432J)** Be an expert user experience researcher with a proven track record of doing research that impacts a complex and diverse product. Position requires occasional travel to unanticipated locations.

Openings in **Seattle, WA (multiple openings/various levels)**:

Product Security Engineer (2317J) Provide security guidance on a constant stream of new products and technologies.

Openings in **New York, NY (multiple openings/various levels)**:

Software Engineer (NYSWEB1116J) Create web and/or mobile applications that reach over one billion people & build high volume servers to support our content. Bachelor's degree required. **Software Engineer (NYSWEM1116J)** Create web and/or mobile applications that reach over one billion people & build high-volume servers to support our content, utilizing graduate level knowledge. Master's degree required.

Mail resume to: Facebook, Inc. Attn: SB-GIM, 1 Hacker Way, Menlo Park, CA 94025. Must reference job title & job# shown above, when applying.