COMPUTING

edge

> Startup Creation

> New 3D Realities

> The Semantic Web

FEBRUARY 2015

www.computer.org

IEEE

IEEE computer society

COMPUTING edge

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

IEEE

IEEE computer society

SUSTAINABLE FORESTRY INITIATIVE Certified Chain of Custody At Least 25% Certified Forest Content www.sfiprogram.org SFI-01042

Printed with inks containing soy and/or vegetable oils

STAFF

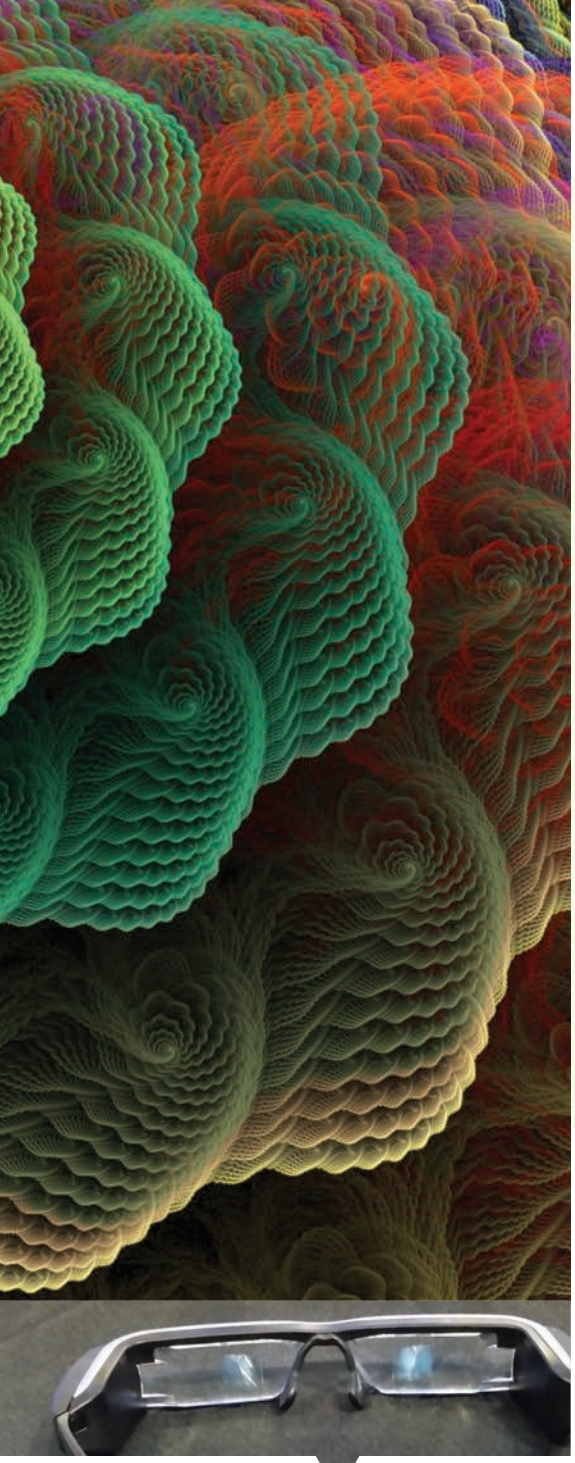
BPA WORLDWIDE

Circulation: ComputingEdge 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 undelivered copies and address changes to IEEE Membership Processing Dept., 445 Hoes Lane, Piscataway, NJ 08855. Application to Mail at Periodicals Postage Prices is pending at New York, New York, and at additional mailing offices. Canadian GST #125634188. Canada Post Corporation (Canadian distribution) publications mail agreement number 40013885. Return undeliverable Canadian addresses to PO Box 122, Niagara Falls, ON L2E 6S8 Canada. Printed in USA.

FABRUARY 2015 • VOLUME 1, NUMBER 2

COMPUTING edge



60
Seeing Is Believing

Departments

NOTE TO READERS

Tech Startups: The Next Big Thing

In the last six months, three different colleagues have left secure jobs to join friends creating tech startup ventures—trading income security and paid benefits (and the nine-to-five grind) for greater autonomy, creativity, and a chance to catch the "next big thing."

Despite the daunting fact that 75 percent of startups fail—25 percent within the first year—the allure of becoming the next Reddit or Dropbox or Airbnb attracts more than half a million fledging tech business entrepreneurs, not to mention countless investors, every year.

This month *ComputingEdge* presents eight articles of interest to startup principals, investors, and consultants—and anyone else keeping an eye on this growing trend.

First *Computing Now*'s editorial content manager Margo McCall offers an update on how Congress's 2012 JOBS Act is affecting crowdfunding for startups. Then, from *IEEE Software* comes a recent Voice of Evidence column investigating what we know about software development in startups. And *Computer*'s legal columnist Brian Gaff suggests some important intellectual property issues anyone contemplating a startup should consider.

Other pertinent articles from *Computing in Science & Engineering*, *IEEE Micro*, and *IEEE Cloud Computing* survey vertical integration (important to any startup expansion), variety and diversity as these values contribute to entrepreneurship, and the Federal Risk Authorization and Management Program (FedRAMP) that's intended to standardize security protocols for cloud service providers and benefit clients of all sizes.

Finally come two "outlier" perspectives: Jason Bay writing in *Computer* describes how his free web resource for students interested in video game development morphed into a lucrative ad site, and an *IT Pro* interview with Microsoft's CIO Jim Dubois provides an insider's perspective on innovation, both large scale and small.

As a startup itself, *ComputingEdge* shares this spirit of adventure—and the hope for prosperity too.

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

Published by the IEEE Computer Society

Magazine Roundup

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

Computer

The February 2015 issue of our flagship publication, *Computer*, surveys current technological advances poised to revolutionize medicine—creating tremendous opportunities for real-time, patient-personalized monitoring and treatment, but also posing significant risks for medical data security. How do developers negotiate these interests?

IEEE Software

Wireless sensor networks are crucial to linking the many elements that make up the Internet of Things, with benefits from greater physical well-being to a reduced energy footprint. In *IEEE Software*'s January/February 2015 issue, a team from SensorHound and Purdue University propose software tools to enhance such networks' reliability.

IEEE Security & Privacy

Cryptography that keeps data secret underlies all secure Internet communication and access control. *IEEE S&P*'s January/February 2015 issue focuses on current trends in cryptography, a field where increasing demand for secure communication is creating a vibrant, emerging landscape for privacy and authentication.

IEEE Internet Computing

To foil phishing attacks, Internet banking websites use security images as part of the login process. But do users notice when a security image is

missing? A study reported in *IEEE Internet Computing*'s January/February 2015 issue finds that most people enter their password even when a security image and caption aren't present.

IEEE Computer Graphics and Applications

Psychologists often evaluate the regulation strategies people use to control and appropriately express their emotions—particularly in adolescents. *IEEE CG&A*'s January/February 2015 Applications department demonstrates an interactive virtual-reality psychotherapy game to aid in the early detection of dysfunctional emotional regulation.

Computing in Science & Engineering

"If simulation is the third tier of science," write the guest editors of *CiSE*'s January/February 2015 issue, "then the communities that build the simulation software are the engine of innovation." This special issue presents the challenges and collective efforts of scientific software communities.

IEEE Intelligent Systems

A team writing in the November/December 2014 issue of *IEEE Intelligent Systems* suggest that advances in cloud computing, the Internet of Things, human–computer interaction, big data, and other fields are fusing the social, cyber, and physical worlds to create a challenging hyper-world that uses data as a common bridge.

IT Professional

Can malware be exterminated? Pessimists believe complete elimination is impossible; optimists argue for eventual solvability. A feature in *IT Pro*'s November/December 2014 issue reveals some pitfalls in malware research that, if addressed, could help move us in the right direction.

IEEE Pervasive Computing

Most wearable devices are powered by batteries that need frequent recharging—difficult if not impossible in remote areas. Researchers writing in *IEEE Pervasive*'s October–December 2014 issue propose wearable harvesters that can harness human kinetic energy to provide sustainable power levels for wearable computing systems.

IEEE Micro

Component miniaturization in electronic devices has raised considerable reliability issues. Writing in *IEEE Micro*'s November/December 2014 issue, a team from the Nara Institute of Science and Technology propose a low-cost, self-tuning scheme to quickly locate defective processing elements or network connections.

IEEE MultiMedia

Visual tracking has many practical applications, from automated surveillance to medical imaging, but remains challenging. *IEEE MultiMedia*'s October–December 2014 issue presents a new model of structured representation for visual object tracking.

IEEE Annals of the History of Computing

The October–December 2014 special issue of *IEEE Annals* focuses on the Algol language, dating from 1958, that's widely considered a major turning point in the development of programming languages and of software in general.

IEEE Cloud Computing

Traditional security mechanisms tailored for small-scale data don't meet the needs of big data analytics and storage applications. The September 2014 issue of *IEEE Cloud Computing* offers innovative security and privacy mechanisms for big data applications in a cloud environment.

Computing Now

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

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

Carrying out Phylogenetic Analyses through Computational Model Checking

Ying Xu, University of Georgia

This installment highlighting the work published in IEEE Computer Society journals comes from IEEE/ACM Transactions on Computational Biology and Bioinformatics.

Phylogenetic analysis is the main technique used to study evolutionary relationships among a given collection of organisms or homologous biomolecules, such as DNA and proteins encoded or used in these organisms. Researchers use such analyses to classify specified organisms taxonomically, determine the origins of protein-encoding genes, and infer how proteins function based on their conserved sequence motifs.

Typically, phylogenic analysis involves two steps: constructing a phylogenetic tree or network based on the given biosequences and deriving biological information from the tree or network. Many computational methods have been developed to construct phylogenetic trees, but in most published studies the biological information from the phylogenetic trees is derived manually.

In "Temporal Logics for Phylogenetic Analysis via Model Checking" (*IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 10, no. 4, 2013, pp. 1058–1070), Jose Ignacio Requeno and his colleagues present a novel way to derive biological information from a phylogenetic tree or network using model-checking techniques. Specifically, the authors represent a phylogenetic tree (or a network, as they plan to do in the future) as a mathematical model using temporal logic and the hypothesized biological properties of the tree.

> Ignacio Requeno and his colleagues present a novel way to derive biological information from a phylogenetic tree using model-checking techniques.

The authors propose a computational scheme using tools established in the fields of temporal logic and model checking to computationally determine whether the represented phylogenetic tree has the predicted properties, saving evolutionary biologists the time and effort required to manually examine the hypothesized properties against large quantities of biological data. While the presentation may be somewhat technical for readers who aren't specialists, particularly in biology, the paper clearly outlines the overall logic.

The methodology presented in this paper can potentially be applied to other model-checking problems in data-intensive biological fields, including dynamic property analyses such as flux analyses over provided metabolic networks; dynamic property studies of biomolecular structures such as protein, DNA, or RNA structures; and inferring the structural properties of a collection of given genomes.

This work represents a new and exciting development linking fundamental computer science theories with data-intensive and increasingly model-intensive fields of modern biology. Bringing these two fields together should yield healthy advances in both. C

YIN XU is a professor in the Department of Biochemistry and Molecular Biology at the University of Georgia. Contact him at xyn@bmb.uga.edu.

Entrepreneurs and Industry Await New Crowdfunding Rules

Margo McCall

If you've been delaying your tech startup until crowdfunding takes off, you may have to wait a little longer.

Congress's passage in 2012 of the JOBS ("Jumpstart Our Business Startups") Act, with overwhelming bipartisan support, was heralded as a historic victory for entrepreneurs. No longer exclusively dependent on Wall Street for funding, startups could reach out to regular people as investors. But two years later, the US Securities and Exchange Commission has yet to finalize rules on two of the act's key provisions.

Titles III and IV of the act would let startups raise up to $1 million from unaccredited investors and up to $50 million through solicitations. The SEC initially expected approval of both rules by the end of last year. But in December, the commission pushed that date up to October 2015, with the rules not taking effect until early 2016.

Crowdfunding Industry on Hold

Meanwhile, an entire industry including platform creators, financial advisors, law firms, consultants, and trade associations—not to mention capital-hungry startups and their founders—waits in virtual limbo for crowdfunding's promise of fundraising democratization to take hold.

"The regulators don't know yet how to manage it and control it. There's a fear of the unknown," says RocketHub CEO Brian Meece, a scheduled speaker at IEEE Computer Society's Startup Rock Stars event, taking place on March 24th in San Francisco. "It's a very tricky thing to get right. We'd rather the regulations be right than fast."

The JOBS Act was intended to help alleviate the funding gap and regulatory concerns startups and small businesses face in connection with raising relatively small amounts of capital.

 Published by the IEEE Computer Society

Leonhardt Ventures Executive Chairman Howard Leonhardt, another Startup Rock Stars speaker, lobbied for passage of the JOBS Act as Startup California's state spokesperson. He's given more than 40 speeches promoting it.

Leonhardt says he became interested in financial reform after receiving bills totaling $4.3 million in underwriting fees, commissions, and road show costs following his biotech company Bioheart's February 2008 initial public offering—more than twice what the IPO netted.

Protecting Future Investors

SEC Chair Mary Jo White has noted that the JOBS Act's intent was to make it easier for startups and small businesses to raise capital from a wide range of potential investors, as well as to provide new opportunities for investors. But it's also important that this market "thrive in a safe manner for investors," she added.

The proposed rules would let companies raise up to $1 million per year through crowdfunding offerings. Individuals could invest up to $100,000 in such offerings, but would have to hold onto their investments for at least a year.

In addition, companies would have to disclose information about their officers and owners and also supply financial reports. Companies outside the US already reporting to the SEC or that have no business plan would be ineligible.

To protect investors, Title III requires that broker-dealers and funding portals be registered with the SEC, provide investors with educational materials, and take measures to reduce the risk of fraud. The portals would be prohibited from handling funds or offering investment advice.

But while Title III's goal was to give startups a less expensive way to raise money, a *New York Times* article estimated that complying with the new SEC regulations would cost companies a lot: about $6,500 for the offering, $4,000 for each annual report, and $28,700 a year for companies required to submit audited financials. Funding intermediaries can also charge a commission between 5 and 15 percent.

Reward versus Equity Crowdfunding

Title II of the JOBS Act, which became law in September 2013, allowed private companies to solicit investment and use reward and donation crowdfunding sites such as Kickstarter and Indiegogo to raise money—which was in itself a major victory for crowdfunding.

Globally, reward and donation crowdfunding raised more than $5 billion in 2013. Reward crowdfunding is used to raise small amounts of money from large numbers of people for everything from disaster relief, charities, and artists' projects to new software and high-tech products. Increasingly, it's viewed as a way to test interest in new products and projects and generate awareness before getting larger investors involved.

Those who donate can receive benefits such as meetings with company founders or product samples, but they can't receive any shares in the company. Currently, an offer or sale of securities must be registered with the SEC unless an exemption such as Title III is available.

As the National Crowdfunding Association describes it, most entrepreneurs start their business by maxing out credit cards then getting "friends and family" financing, before moving on to "angel" financing from a wealthy individual, bank loans, and money from venture capitalists. Crowdfunding expands the "friends and family" stage, according to the association, and it's often a step on the path to getting angel and venture capital investors.

According to Meece, though, the jury's still out on the JOBS Act overall: "We'll have to wait a while to see ultimately what happens."

Margo McCall is editorial content manager for the Computer Society's Computing Now destination site (http://computingnow.computer.org). Have an idea for an article or wish to contribute? Contact Margo at mmccall@computer.org.

Editor: Tore Dybå
SINTEF
tore.dyba@sintef.no

What Do We Know about Software Development in Startups?

Carmine Giardino, Michael Unterkalmsteiner, Nicolò Paternoster, Tony Gorschek, and Pekka Abrahamsson

STARTUPS ARE NEWLY created companies with little or no history of facing high volatility in technologies and markets. In the US alone, 476,000 new businesses are established each month,[1] accounting for nearly 20 percent of job creation.[2] As such, startups are an important factor in the economy. However, the environment of startups is dynamic, unpredictable, and even chaotic, forcing entrepreneurs to act quickly, fail fast, and learn faster to find a market niche and acquire a sustainable income. Sixty percent of startups don't survive the first five years, and 75 percent of venture capital funded startups fail.[3] Most of this is due to the high risk of startups, missed market windows, and other business reasons. To what extend engineering practices impact this high failure rate is still unknown given the premature state of research.

We present a detailed investigation and collection of all known empirical software engineering sources related to startups and their engineering practices, as well as an analysis of how accurate and reliable this available evidence is.[4] We see this as a first critical step into a largely unknown area—the world of software engineering practices in startups.

> High uncertainty and rapid evolution are the two key characteristics for startups.

What Is a Startup, Anyway?

In the past, the term "startup" had different meanings. Looking at the recurrent themes (Table 1 offers a complete list) adopted by researchers and practitioners, a startup is a small company exploring new business opportunities, working to solve a problem where the solution isn't well known and the market is highly volatile. Being newly founded does not in itself make a company a startup. High uncertainty and rapid evolution are the two key characteristics for startups retrieved by the studies, which better differentiate them from more established companies.

We retrieved and evaluated empirical evidence by using the systematic mapping study approach (see the sidebar).

Startup Software Development

"Done is better than perfect" and "move fast and break things" are slogans you might read when entering a startup workspace. What stands behind those slogans is a summary of more than 200 working practices. We reviewed these to point out where gaps exist and future development and research are warranted.

Process Management Is Agile, Evolutionary, and Opportunistic

Process management represents all the engineering activities used to manage product development in startups. Because the flexibility to accommodate frequent changes is essential in the startup context, agile

 Published by the IEEE Computer Society

TABLE 1

Recurrent themes in software startups.

Theme	Description
Lack of resources	Economical, human, and physical resources are extremely limited.
Highly reactive	Startups are able to quickly react to changes in the underlying market, technologies, and product (compared to more established companies).
Innovation	Given the highly competitive ecosystem, startups need to focus on and explore highly innovative segments of the market.
Uncertainty	Startups deal with a highly uncertain ecosystem under different perspectives: market, product features, competition, people, and finance.
Rapidly evolving	Successful startups aim to grow and scale rapidly.
Time pressure	The environment often forces startups to release fast and to work under constant pressure (terms sheets, demo days, investors' requests).
Third-party dependency	Due to lack of resources, startups heavily rely on external solutions to build their product: external APIs, open source software, outsourcing, COTS, and so on.
Small team	Startups start with a small number of individuals.
One product	Company activities gravitate around one product/service only.
Low-experienced team	A good part of the development team is formed by people with less than five years of experience and often recently graduated students.
New company	The company has been recently created.
Full organization	Startups are usually founder-centric, and everyone in the company has big responsibilities, with no need for upper management.
Highly risky	The failure rate of startups is extremely high.
Not self-sustained	Especially in the early stage, startups need external funding to sustain their activities (venture capitalist, angel investments, personal funds, and so on).
Little working experience	The basis of an organizational culture isn't present initially.

methodologies have been considered the most viable process—they embrace change, allowing development to adapt to the business strategy.[5] Fast release with an iterative and incremental approach shortens the lead time from idea conception to production with fast deployment.

A variant to agile is the lean methodology,[6] which advocates the identification of the riskiest parts of a software business and provides a minimum viable product to systematically test and plan modification for the next iteration. In this regard, prototyping is essential to shorten the time to market.

To allow better prototyping activities, evolutionary workflows are needed to implement "soft-coded" solutions in the first phases until the optimal solution is found. Despite the number of methodologies that embrace fast prototyping in development, none of the processes are strictly followed by startups. Yet, the uncertainty and fast-changing needs of startups drive them to opportunistically tailor minimal process management to their short-term objectives and adapt to the fast-paced learning process of their users to address market uncertainty.

Software Development Is Driven by Customers who Act as Designers

Startups are under constant pressure to rapidly demonstrate that they're developing a solution that fixes a real problem.[7] They're constantly optimizing the problem/solution fit. To achieve it, startups must discover the real needs of their first customers, testing business speculations

EMPIRICAL BODY OF EVIDENCE

A systematic mapping study is a method to structure the empirical evidence in a particular field of interest.[1] We identified 43 studies that investigate different aspects of startups and their software development processes. We also estimated the strength of evidence in this field by assessing the rigor and relevance of the studies (see Figure A).[2] *Rigor* refers to the precision and thoroughness of reporting a study's design, validity threats, and results. *Relevance* refers to the realism of the environment in which the study is performed and to the potential of transferring results to practitioners.

Our rigor and relevance assessment suggests that the empirical evidence on the startup phenomenon is still rather premature. A minority—10 of the 43 mapped studies—provides transferable and reliable results to practitioners (sector A). Similarly, 10 studies provide low rigor and relevance (sector C). More studies (23) exhibit moderate industry relevance, but with low scientific rigor (sector B). From this observation, we conclude that it's challenging to conduct research in an environment in which a lack of resources is a dominant characteristic. Researchers need to identify efficient means to collaborate with and study startups.

References

1. K. Petersen et al., "Systematic Mapping Studies in Software Engineering," *Proc. 12th Int'l Conf. Evaluation and Assessment in Software Eng.* (EASE), 2007, pp. 1–10.
2. M. Ivarsson and T. Gorschek, "A Method for Evaluating Rigor and Industrial Relevance of Technology Evaluations," *Empirical Software Eng.*, vol. 16, no. 3, 2010, pp. 365–395.

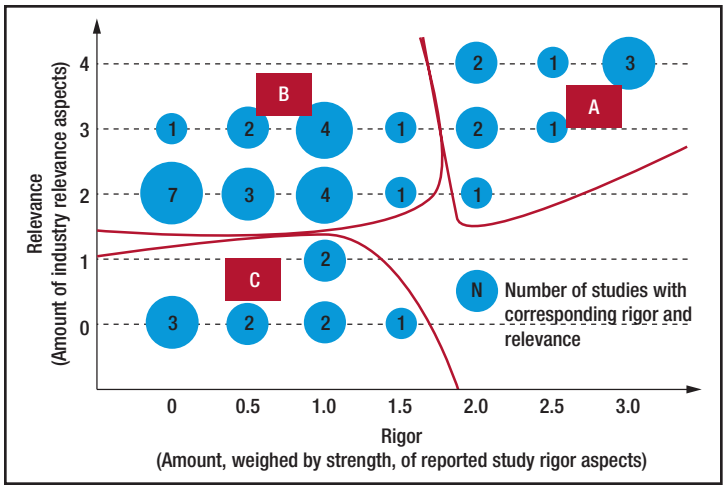


FIGURE A. Rigor and relevance of primary studies.

only by defining a minimal set of functional requirements.[8]

Several authors acknowledge the importance of involving the customer/user in the process of eliciting and prioritizing requirements according to their primary needs. However, the market-driven nature of those requirements also demands alternatives. For example, startups can use scenarios to identify requirements in the form of user stories and estimate the effort for each story. However, polishing requirements that address an unsolicited need a waste of effort. Requirements elicitation methods are moving toward testing the problem and understanding if the solution fits real needs before the product goes to market (the so-called customer development process).[7]

In the startup context, customers often steer requirements, and developers must be ready to embrace change from day one. The use of architecture and design patterns to make features modular and independent is crucial when functionality is continuously updated or removed. Therefore, employing architectural practices and frameworks that enable easy extension of the design can dramatically benefit the alignment between the product and market uncertainty.[9] This requires some upfront effort but can prevent the growth of product complexity.

Scientific evidence also points to the advantages of constant code refactoring. Reimplementing the whole system might be costly and risky if it must be immediately scalable to a growing number of users. Therefore, some quality assurance is needed for the functionality that brings the most value to customers. The use of ongoing customer acceptance through focus groups made up of early adopters can provide a time-efficient way to discover major bugs. But solutions are still scarce for easily accessible automated testing frameworks and the more practical user interface-testing approaches.

The Team Is the Catalyst of Development

Time pressure and lack of resources often lead startups to adopt a loose organizational structure without traditional management hierarchies.[10]

Empowerment of team members represents the main viable strategy for enhancing performance and success.[11] The team must be able to absorb and learn from trial and error quickly enough to adapt to new emergent practices. Working on innovative products requires creativity—an ability to adapt to new roles and face new challenges every day, working overtime if necessary.

Indeed, in building a startup company, the team needs expertise to counterbalance its lack of resources. In addition, having previous experience in similar business domains and exhibiting entrepreneurial characteristics (courage, enthusiasm, commitment, leadership) are important parts of a startup employee's skillset.

Nevertheless, the absence of structure might hinder important activities, such as sharing knowledge and team coordination, especially when the company grows. In this case, collocation is essential to facilitate informal communication and close interactions between team members.

Tools Can Accommodate Product and Management Changes

Startups can take advantage of the newest technologies and development tools without having to worry about legacy or previous working experiences.[12] But the selection of a technology requires some domain- or product-specific requirements, which are typically unknown in the early stages.

In general, startup employees prefer using those technologies that can quickly accommodate change in the product and its management.[13] Examples include general-purpose infrastructures, such as configuration management, problem reporting, tracking, and planning systems, and scheduling and notification systems. Easy-to-implement tools, such as whiteboards and technologies that can handle fast-paced changing information, will lower a startup's training and maintenance costs. To mitigate the lack of resources, startups often appear to take advantage of open source solutions when possible, which also give them access to a large pool of evaluators and evolving contributions.

Nevertheless, the absence of structure might hinder important activities.

Startup companies seek to generate revenue and obtain funding to continue the development, which means that software quality isn't their most critical concern. To quickly validate the product, they tend to use agile and lean methods in an ad hoc manner.[14]

Evidence suggests that engineering activities must be tailored to the startup context to allow flexibility and reactiveness in development workflows. Decision makers in startups confront continuous unpredictability; the relationship between cause and effect can only be perceived in retrospect.[15] Applying rigorous methodologies to control development activities isn't effective because no matter how much time is spent on analysis, it isn't possible to identify all the risks or accurately predict what practices are required to develop a product.

On the other hand, flexible and reactive methods designed to stimulate customer feedback increase the number of perspectives and solutions available to decision makers. Developers need the freedom to choose activities quickly, stop immediately when the results are wrong, fix the approach, and learn from previous failures. In line with the lean startup movement, we would expect methodologies and techniques tailored from common agile practices to specific startups' cultures and needs; failures should be completely acceptable or even preferred in favor of a faster learning process.

Reported common practices, which ride the wave of rapidly evolving technologies and markets, are as follows:

- use of well-known frameworks to quickly change the product according to market needs;
- use of evolutionary prototyping and experimentations via existing components;
- ongoing customer acceptance through early adopters' focus groups;
- continuous value delivery, focusing on core functionalities that engage paying customers;
- empowerment of teams to influence final outcomes;
- use of metrics to quickly learn from consumers' feedback and demand; and
- use of easy-to-implement tools to facilitate product development

and handle fast-paced, changing information.

Today's startups are at the forefront of applying new technologies in practice. The growing startup phenomenon opens uncharted opportunities as well as challenges in research. "Startuppers" need more transferable and reliable results concerning the diversity of context and viewpoints in the adoption of practices dealing with high uncertainty.

References

1. R.W. Fairlie, "Kauffman Index of Entrepreneurial Activity," Kauffman Foundation, 2014.
2. R.W. Fairlie, "State of Entrepreneurship Address," Kauffman Foundation, 2014.
3. C. Nobel, "Why Companies Fail, and How Their Founders Can Bounce Back," Harvard Business School, 2011.
4. N. Paternoster et al., "Software Development in Startup Companies: A Systematic Mapping Study," *Information and Software Technology*, 2014; DOI: 10.1016/j.infsof.2014.04.014
5. G. Coleman and R. O'Connor, "An Investigation into Software Development Process Formation in Software Start-ups," *J. Enterprise Information Management*, vol. 21, no. 6, 2008, pp. 633–648.
6. E. Ries, *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*, Crown Business, 2011.
7. S. Blank, "Why the Lean Start-Up Changes Everything," *Harvard Business Rev.*, vol. 91, no. 5, 2013, p. 64.
8. S.-C. Li, "The Role of Value Proposition and Value Co-Production in New Internet Startups: How New Venture e-Businesses Achieve Competitive Advantage," Portland Int'l Center for Management of Engineering and Technology (PICMET), 2007, pp. 1126 –1132.
9. S. Yogendra, "Aligning Business and Technology Strategies: A Comparison of Established and Start-up Business Contexts," *Proc. Int'l Eng. Management Conf.* (IEMC), 2002, pp. 2–7.
10. Y.-W. Yu et al., "Entrepreneurial Success for High-Tech Start-ups: Case Study of Taiwan High-Tech Companies," *Proc. 6th Int'l Conf. Innovative Mobile and Internet Services in Ubiquitous Computing*, 2012, pp. 933–937.
11. E. Carmel, "Time-to-Completion in Software Package Startups," *Proc. 27th Hawaii Int'l Conf. System Sciences*, 1994, pp. 498–507.
12. S.M. Sutton, "The Role of Process in Software Start-ups," *IEEE Software*, vol. 17, no. 4, 2000, pp. 33–39.
13. M. Crowne, "Why Software Product Start-ups Fail and What to Do about It," *Proc. Int'l Eng. Management Conf.* (IEMC), 2002, pp. 338–343.
14. S.W. Ambler, "Lessons in Agility from Internet-Based Development," *IEEE Software*, vol. 19, no. 2, 2002, 66–73.
15. C.F. Kurtz and D.J. Snowden, "The New Dynamics of Strategy: Sense-Making in a Complex and Complicated World," *IBM Systems J.*, vol. 42, no. 3, 2003, pp. 462–483.

CARMINE GIARDINO is a PhD student in computer science at the Free University of Bolzano. His research interests include software development in startup companies, focusing on the alignment between business strategies and development activities. Contact him at cgiardino@unibz.it.

MICHAEL UNTERKALMSTEINER is a PhD student in software engineering at the Blekinge Institute of Technology (BTH). His research interests include coordination of requirements engineering and testing, information retrieval to support decision making, and software repository mining. Unterkalmsteiner received an MSc in software engineering from BTH. Contact him at mun@bth.se.

NICOLÒ PATERNOSTER is a startupper at www.woodwallets.io. His research interests include bitcoins service development and makerspace development. Paternoster received an MSc in software engineering from Blekinge Institute of Technology and Free University of Bozen-Bolzano. Contact him at hi@adva.io.

TONY GORSCHEK is a professor at the Blekinge Institute of Technology (BTH) and Chalmers. His research interests include requirements engineering, technology and product management, process assessment and improvement, and practical innovation. Gorschek received a PhD in software engineering from BTH. Contact him at tgo@bth.se.

PEKKA ABRAHAMSSON is a full professor of computer science at Free University of Bozen-Bolzano. His research interests include empirical software engineering, agile development, startups, and cloud computing. Abrahamsson received a PhD in software engineering from University of Oulu in Finland. Contact him at pekka.abrahamsson@unibz.it.

This article originally appeared in IEEE Software, *vol. 31, no. 5, 2014.*

Intuit, Inc. has openings for the following positions (by location):

San Mateo County, including **Menlo Park**; Santa Clara County, including **Mountain View**: **Software Engineers (Job code: G1):** Design, develop, troubleshoot and/or test/QA software. **Senior Software Engineers (Job code: G2):** Use knowledge of software engineering best practices and principles to design and develop web applications. **Staff Software Engineers (Job code: G3):** Use technical expertise to develop code and unit test for software and/or analyze user needs and/or software requirements to determine required software improvements and/or modifications. **Software Engineers in Quality (Job code: G4):** Design, create, document, and/or implement test strategies, test automation and quality tools and processes to ensure quality of products and services. **Senior Software Engineers in Quality (Job code: G5):** Use knowledge of software engineering best practices and principals to design, create, document, implement and/or maintain test scripts for complex on-demand and integration applications. **Senior Applications Operations Engineers (Job code: I-370)**: Drive the design, development and implementation of operational standards and capabilities for connected services. **Online Acquisition Marketers (Job code: I-7):** Serve as the Online Acquisition Lead for QuickBooks Ecosystem Creative to be responsible for the development of an OA creative brief for the QB Ecosystem and coordinate the relationship with our external agency partner. **Senior Business Analysts (Job code: I-65):** Partner closely with product and marketing managers to help guide strategic decision making on product and marketing tactics/strategy using data. **Business Data Analysts (Job code: I-168):** Interpret large volumes of data to tease out actionable insights, telling a story that drives revenue, product and/or business change. **Data Engineers (Job code: I-45):** Responsible for the design, development, and implementation of data movement and integration processes in preparation for analysis, data warehousing, and operational data stores involving very large quantities of data. **Group Managers (Job code: I-288):** Define the roadmap to achieve strategies that will drive quality product experiences for customers and will accelerate business growth. **Development Managers (Job code: I-346):** Supervise and contribute to the design, development, testing, and deployment of web-based applications. **Sr. Product Managers (I-460):** Identify deep customer insights that lead to better products and marketing/messaging methods. **Sr. Product Managers (Job code: I-315):** Lead innovation in products and business models, primarily in the areas of Small Business Accounting, Payments, Point of Sale and QuickBooks ecosystem offerings. May require up to 20% international travel. **Senior Technical Data Analysts (Job code: I-105):** Engage with key stakeholders to understand critical business requirements and identify ways that analytics can best support or optimize business growth. Access and synthesize data using appropriate tools and technology. **San Francisco, California: Staff Software Engineers (Job code: G3-SF):** Use technical expertise to develop code and unit test for software and/or analyze user needs and/or software requirements to determine required software improvements and/or modifications. **Senior Software Engineers in Quality (Job code: G5-SF):** Use knowledge of software engineering best practices and principals to design, create, document, implement and/or maintain test scripts for complex on-demand and integration applications. **San Diego, California: Software Engineers (Job code: G1-SD):** Design, develop, troubleshoot and/or test/QA software. **Senior Software Engineers (Job code: G2-SD):** Use knowledge of software engineering best practices and principles to design and develop web applications. **Staff Software Engineers (Job code G3-SD):** Use technical expertise to develop code and unit test for software and/or analyze user needs and/or software requirements to determine required software improvements and/or modifications. **Software Engineers in Quality (Job code: G4-SD):** Design, create, document, and/or implement test strategies, test automation and quality tools and processes to ensure quality of products and services. **Senior Software Engineers in Quality (Job code: G5-SD):** Use knowledge of software engineering best practices and principals to design, create, document, implement and/or maintain test scripts for complex on-demand and integration applications. **Staff Application Operation Engineers (Job code: I-362):** Drive the design, development and implementation of operational standards and capabilities for connected services that enable highly available, scalable & reliable customer experiences. **Senior Systems Engineers (Job code: I-38):** Consult with business unit partners to define application requirements for computing, storage and networking. **Woodland Hills, California: Staff Software Engineers (Job code: G3-LA):** Use technical expertise to develop code and unit test for software and/or analyze user needs and/or software requirements to determine required software improvements and/or modifications. **Staff Data Engineers: (Job code: I-107):** Design, develop, and implement data movement and integration processes in preparation for analysis, data warehousing, or operational data stores, involving very large quantities of data. **Reno, Nevada: Software Engineers (Job code: G1-NV):** Design, develop, troubleshoot and/or test/QA software. **Plano, Texas: Software Engineers (Job code: G1-TX):** Design, develop, troubleshoot and/or test/QA software. **Senior Software Engineers in Quality (Job code: G5-TX):** Use knowledge of software engineering best practices and principals to design, create, document, implement and/or maintain test scripts for complex on-demand and integration applications. **Senior Systems Engineers (Job code: I-172):** Serve as a core member of IT support team charged with the operations of infrastructure systems primarily providing monitoring and management capabilities for IT and application operations teams. **PTG Analytics Leaders (Job code: I-103):** Lead and develop a team of business analysts as well as integrate deeply in the business and provide timely and effective insight as a trusted business partner. **Cambridge, Massachusetts: Software Engineers (Job code: G1-MA):** Design, develop, troubleshoot and/or test/QA software. **Staff Business Analysts (Job code: I-382):** Manage cross-functional teams to define, build, and implement business process and technology solutions that increase efficiencies, improve decision support/analytics capabilities. Requires 5% domestic travel.

Submit resume to Intuit Inc., Attn: Olivia Sawyer, J203-6, 2800 E. Commerce Center Place, Tucson, AZ 85706.
You must include the job code on your resume/cover letter. Intuit supports workforce diversity.

IP Issues for Startups

Brian M. Gaff, McDermott Will & Emery, LLP

Most startups face myriad issues, even before opening their doors. This includes getting financing, hiring the right people, finding office space, and—importantly—getting the company's intellectual property under control and adequately protected.

If you're contemplating starting a new company, there are intellectual property (IP) issues that you'll need to address. Many of these are complex, and most should be evaluated and acted upon only after receiving competent legal advice that's specific to your situation. However, as a starting point, you should pay attention to two major issues: freedom to operate and the initial IP strategy.

For an expanded discussion on this topic, listen to the podcast that accompanies this column at www.computer.org/computing-and-the-law.

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

FREEDOM TO OPERATE

Freedom to operate is a straightforward concept that's often overlooked. Simply put, it means verifying that you'll have the ability to offer your product or service as you envisioned, including whether you can use the name you've chosen for your product or company. This affects marketing and your planned online presence, particularly if the domain names you've chosen are unavailable.

More critically, you need to ensure that there aren't patents that others own that might cover your product or service. Such patents will need to be scrutinized—ideally, well before your launch. Otherwise, you might find out that you're infringing on those patents and could be prevented from making, offering to sell, selling, using, or importing your product or service.

Commissioning a freedom to operate search is one way to avoid unpleasant surprises. This involves searching for patents and trademarks that cover subject matter that's similar to yours. This can be done on your own, or with the help of search firms or lawyers.

One of the first things to consider before starting the search is determining where to market your product or service. This is because most IP protections like patents and trademarks don't have any effect beyond the borders of the countries in which they're granted. In other words,

 Published by the IEEE Computer Society

EDITOR BRIAN M. GAFF
McDermott Will & Emery, LLP; bgaff@mwe.com

if you intend to market your product in the US, then you need to search for related US patents and trademarks. If you plan to market in Canada, then look for Canadian patents and trademarks.

Once you determine which countries to search, you should then start looking for trademarks that might be similar to the names you've chosen for your product or company. For US trademarks, this means going to the US Patent and Trademark Office website (www.uspto.gov) and navigating to the trademark search area. Simply searching for your product or company names should provide initial results on whether the names are already in use or similar to other names that are already in use. Names that are in use are "registered."

In many instances, interpreting these results isn't easy. For example, similar names for similar products that are already registered might cause a problem that could lead to a trademark infringement suit. In rare cases, and when it's clear that you've done a correct and comprehensive search, you might get results that show your names are unique and unlikely to conflict with someone else's trademarks. However, for most people, the better approach is to have a trademark professional handle the search, evaluate the results, and give you a recommendation. Although this research has a cost, it's well worth it considering what's at stake. If you don't get "clearance" on your names, you might lose your entire marketing strategy and corporate identity in a flurry of infringement allegations.

A freedom to operate analysis of patents is typically more complicated than a trademark analysis. It can start the same way—you can search the patent offices of the countries of interest for technologies that are similar to yours. A goal is to find issued patents that have patent claims that might cover your product. This isn't an easy task, especially if you're unfamiliar with patent searching and claim interpretation. That's why most people have others handle this for them.

An efficient search process starts with preparing an accurate and concise description of the product or service that you intend to sell. One alternative is to provide this description to a patent-searching firm that will navigate through a multitude of categories of patented inventions to find similar subject matter. Low-cost searches—those that are under US$1,000—are typically limited to just a few hours to perform the search and don't involve extensive analysis of the patent claims. However, this might be adequate in some cases.

At the higher end are searches performed by patent attorneys. These searches can start the same way, with a description of the product or service. They might even involve getting initial results from a search firm, but this is just the initial phase. For example, the patent attorney might follow up with the search firm to perform refined searches based on the initial results. More importantly, however, the patent attorney will carefully analyze the claims in the patents, using the proper legal methods to determine what they cover. This analysis is usually complex and time-consuming and, in many cases, the results are presented in a formal, written legal document known as an *opinion*.

An opinion of this type is usually prepared for the technology that's most critical for the start-up—the so-called "crown jewels." A less extensive approach might be appropriate for secondary technologies. You should determine where to draw the line and what type of analysis to perform only after adequate consultation with your lawyer.

INITIAL IP STRATEGY

Assuming that you've received the clearances needed from your freedom to operate analyses, the next step is to develop an IP strategy to use during the start-up process and for the initial stage of the company. This strategy should evolve over time based on the company's direction and needs.

One of the first priorities is to protect the identities of the start-up and the product. That means applying for one or more trademarks that cover the start-up's name, the name of the product or service, any slogans or taglines, and the domain name. It's appropriate to apply for trademarks in every country where you plan to market. Getting your trademarks registered should be straightforward if you've previously cleared them in your freedom to operate analysis.

If you don't get "clearance" on your names, you might lose your entire marketing strategy and corporate identity in a flurry of infringement allegations.

You should secure copyright protection as well. This can protect your software code, marketing materials, and your website. In the US, it's easy and relatively inexpensive to apply for a copyright. Although copyright-protection can be narrow—it protects against unauthorized copying—it's cost-effective and worth having. For certain types of copyright infringement, the damages awarded to copyright owners are set by statute and can be significant.

Obtaining initial patent protection involves weighing several factors. For example, the expense of preparing and applying for a patent can be significant and might be beyond what the start-up can afford. After determining your budget and conferring with your lawyer, you should decide whether to proceed with a full-fledged patent application—called a *utility application* in the US—or a *provisional patent application*. The provisional option in the US is an attractive one when funds are limited: for a small government fee, an inventor can submit a description of his or her invention to the US Patent Office. The submission is much less formal compared to what's required for a utility application. The tradeoff, however, is that a provisional application never matures into a patent and can't be used to stop someone else from using the technology. A provisional application lasts for only one year and then expires.

A provisional application also allows the applicant to use the one-year period to prove the technology and raise funds. However, the applicant must file a utility application on the same technology, referencing the provisional application, before the latter's expiration. Having the extra time to raise funds might make it feasible to file the utility application.

Whether you decide on a provisional or utility application, it's important that you file it as soon as possible before going public with your product. This is imperative because of the first-to-file patent system in the US. It's especially critical if you plan to pursue patent protection outside the US, because in many other countries patent applicants must show that their inventions have "absolute novelty." In other words, to qualify for a patent, an invention must never have been publicly disclosed. Selling an invention or otherwise disclosing it can harm your chances of obtaining a non-US patent.

Another type of patent to consider is a *design patent*. In the US, design patents cover the ornamental appearance of an object: they don't cover any functional aspects of the object. If your product has a unique or distinguishing appearance that you want to protect, you should file an application for a design patent. Having a design patent can discourage others who might try to copy your product's appearance.

Whether you decide on a provisional or utility application, it's important to file it before going public with your product.

THE UNEXPECTED ROADBLOCK

Let's assume that you've ordered your freedom to operate searches and reviewed the results, and those results identify a problem. If it's a trademark issue—maybe a similar name that's already in use—you'll need to evaluate whether you can proceed using a different name. This will probably provide a quick resolution, but it could be difficult in practice because you'll likely need to make comprehensive changes to your marketing plan. If you can't change the name, another option is to license or purchase the trademark from its current owner. However, that might not be possible in all situations.

Discovering one or more patents that might block your product or service is a more complex problem. Issued US patents are presumed by statute to be valid and enforceable. Therefore, one way to address this problem is to investigate the possibility that the patents are invalid. This is usually done by having a patent lawyer examine the prior art—any materials that predate the patents—and provide a legal opinion on the validity of the patents in view of the prior art. Your lawyer can also prepare an opinion on whether there is infringement. Based on those opinions, you can decide whether to proceed as you originally intended.

Alternatively, and if it's feasible to do so, you can redesign parts of your product or service to differentiate them from the blocking patents. This is called *design around*. This process is usually one where the engineers and patent lawyers work together to ensure that any changes made are sufficient. Certain legal doctrines permit some elasticity in the scope of patent claims, so the lawyer's input is necessary to ensure that the changes have adequately distinguished your product or service from the claims in terms of how a court could interpret those claims.

In the excitement of launching a start-up, certain issues can slip through the cracks. IP issues shouldn't fall victim to that fate. Work with your lawyer from the start to ensure that you've uncovered clearance issues and taken steps to address them. Apply for IP protection early on. Otherwise, you might encounter an eleventh-hour difficulty that derails your ability to market your product as you intended.

This article originally appeared in Computer, *vol. 48, no.1, 2015.*

BRIAN M. GAFF is a senior member of IEEE and a partner at the McDermott Will & Emery, LLP, law firm. Contact him at bgaff@mwe.com.

Enough Variety and Diversity?

Shane Greenstein
Northwestern University

Very little activity in economic life benefits from restricting the supply of services to a single firm. Although that might sound abstract, high-technology markets stretch the principle to its limits.

The routine economic argument for more suppliers stresses the direct gains from competition, such as lower prices and more customer orientation in nonprice activities. Another argument for more suppliers stresses the gains from variety and diversity. This second idea rarely gets much attention, so this column focuses on it.

Consumer electronics markets provide the simplest illustration. TV, digital cameras, and camcorders all contain multiple segments of price and quality. No single firm best supplies all segments, and users would lose from artificially restricting supply to one firm. Being open to more than one firm generates more diversity of models for distinct tastes.

Are there any other arguments for supporting diversity of suppliers in technology markets beyond better pricing and proliferation of models? Anything else?

Avoiding errors

Another benefit from diversity could be labeled "benchmarking." Including a variety of firms lets investors check whether a firm has made an error. Benchmarking is valuable when trying to discern whether a firm has overcommitted to one technological forecast. That type of error is likely when executives grow out of touch with frontier developments, but it also happens to the best CEOs. Markets are complex, and CEOs are just human.

Bill Gates was one of the best CEOs in the history of tech, and his biggest strategic error serves as the canonical illustration. Prior to the spring of 1995, Gates understood how the Internet worked on a technical level, but he misperceived its commercial potential, even as late as early 1995. Gates had not authorized much investment in the Internet, and did not intend to have Microsoft build any applications.

Gates would not have changed his view had there not been a firm with a distinct commercial outlook—Netscape, in this case. Gates had argued against investing, and inside his own firm he had won all the abstract arguments (as he usually did). Netscape's rise gave Gates's employees a concrete and uncomfortable counterexample. Once confronted, Gates changed his view.

The generality about diversity's role should be stated carefully. In high-tech markets, new opportunities can create genuinely different opinions about the appropriate actions to take. When participants perceive the value of opportunities in different ways, there might be no natural way to settle disagreement about which perception is right. That remains so even with the smartest executives. Diversity of firms helps to bring to reality examples that settle disagreement.

In this view, it is useful for investors and managers to benchmark their views against various perspectives. Diversity can encourage the appearance of perceptions that fall outside of the prevailing view.

Diversity and variety might be beneficial for benchmarking, but it is never clear in advance when that will shape decisions. The benefits are prospective and not concentrated on one beneficiary. No dissenting investor ever takes a stand about the decline of benchmarking at an investor conference. I have never heard of an investor dissent against a merger based on the decline in benchmarking, even though every merger works against diversity, almost by definition.

As a result, diversity emerges through indirect means. For example, through supporting more entrepreneurship, investors support more diversity, even when that is not the stated goal. If enough managers disagree with the leadership, they can break off and find investors to help them form their own firm. If talented and perceptive inventors see an untapped opportunity, they can branch off and gain investors to help them build an application to meet those needs.

Entrepreneurship is not a perfect way get more diversity, to be sure. It takes a combination of technical talent and managerial wisdom to be an entrepreneur, not to mention working capital and a big dose of luck. Going to investors can be arduous, even after the recent explosion in angel investing and crowd investing (albeit those trends have improved things).

 Published by the IEEE Computer Society

Lest that sound too sanguine, other forces in investor communities push against diversity. Investor communities exert pressure for one firm not to fall far behind others, and such pressures tend to be conservative. Investors like to compare firms to a prevailing view about which risks are worth undertaking and which are not. Iconoclastic entrepreneurs tend not to benefit from such scrutiny. Experimenting in a countervailing direction can be perceived as risky. Risk might not be fatal, but it can raise costs in subtle ways.

As an aside, engineers and highly talented management read what analysts write. Although some technical talent just loves working for a rebel, most would happily sacrifice a little rebellion for a larger and more reliable paycheck. For these and related reasons, labor markets tend to erode differences between firms over time.

Supporting variety

Other institutions of modern technology markets have ambiguous effects on diversity. Standards committees can support various perspectives while also reducing diversity by standardizing designs. It is almost an oxymoron: less diversity in standards supports more diversity in applications.

Standards committees perform two distinct functions. The first one works toward technical uniformity in some aspects of a design. Most committees pursue that goal in the service of a second function, to enable various additional designs from commercial participants. A well-designed standard can help specialists invent in all sorts of directions without worrying about coordinating with each other.

There are many examples of these two functions at work. The Internet Engineering Task Force standardized protocols, and that supported the growth of the commercial Internet by fostering specialization within layers, enabling many small actors to participate. Another example is the USB standard, which enabled a range of additional complementary products. A third is IEEE 802.11, also known as Wi-Fi, which lets one set of actors make antennae while another makes receivers, and the two do not have to work together each time the design improves.

Another function of a standards committee is less often acknowledged, and also fosters diverse viewpoints. Standards committees can be a forum for allowing otherwise unheard voices to have influence. That happens if a standards committee has comparatively open participation rules, as the IEEE has, for example. It can enable multiple voices that are otherwise powerless to speak in an established firm.

The history of Wi-Fi also illustrates this point. The original committee involved participants from a mix of large and small firms, as well as small divisions within large firms. More to the point, many of the leaders came from a small division of AT&T, which became Lucent, which had been picked up from the NCR acquisition years earlier. They did not have much status in their own organization. At the IEEE, however, they were able to get their views heard, drive a design process to conclusion, and show others that the design was useful.

To be sure, this observation only goes so far. Most standards committees have their own politics as well, and many "open committees" are not inclusive in practice. In the extreme, such politics can become dysfunctional, and a mismanaged committee is not particularly useful for anything. In other words, openness requires smart management.

Private firms often sponsor standards in the context of developing platforms, and that motivates a final question: are private standards any better than quasi-public groups? Private firms do not make things any less ambiguous. Consider the actions of a platform sponsor—say, Apple. It reserves a set of functions for itself, and standardizes the interface, deliberately letting others explore a range of applications. This is a sensible approach when others have a better idea about which applications will be more valuable—in an iPhone app, say. The oxymoron arises again: Apple's uniformity in its standard fosters diversity of applications.

That also moves the question about diversity to a different place. Different firms take distinct approaches to setting up a platform, and in most nascent markets the right approach will be unknown. Users also benefit from having a diversity of platforms—for example, Android and its cousins.

Alas, that too only goes so far. A platform has little incentive to foster the maximal amount of platform diversity. After all, no platform sponsor has ever sponsored enough diversity to give birth to another platform that undermines its own plans or reduces its own profits.

Where does that leave us? Is more diversity always better? Will markets foster enough diversity? A few general observations emerged from this discussion.

More diversity is better in the face of ambiguity about the best way forward in a commercial market. That generality applies to many technology markets, perhaps even the vast majority. It is also a subtle effect to observe. Not all firms do business the same way; they differ in their governance strategies, designs, and other dimensions of operations, especially when the most valuable approach is far from obvious.

An additional benefit to diversity has to do with benchmarking. Diversity helps discern whether a particular firm made a good choice. This question becomes particularly valuable in high-tech markets where more is unknown about the source of value or the right strategy.

In general, therefore, markets usually benefit from some diversity, but it is hard to believe that markets provide the right amount of diversity for investors or users. If they do so, it is only by accident. MICRO

Shane Greenstein is the Kellogg Chair in Information Technology at the Kellogg School of Management, Northwestern University. Contact him at greenstein@kellogg.northwestern.edu.

Editors: Konrad Hinsen, konrad.hinsen@cnrs-orleans.fr | Matthew Turk, matthewturk@gmail.com

Vertical Integration

Matthew Turk | National Center for Supercomputing Applications

The only Web browser tab I always have open is Gmail. It serves as the center of my digital life. When I receive a new email with a Google Docs document linked in it, there's a preview of that document—when I send one to someone else, Gmail first checks to see if the person receiving it has permission to view it and lets me know if they don't. If I have pictures uploaded to my account, I can insert them into an email without re-uploading. The apps that Google provides share data effectively (thanks to a controversial terms-of-service change from a few years ago) and this enables a lower overhead to deploying new features and disseminating those features to users. My friends that use iOS products find similar smoothness of integration between different apps provided by Apple, but Apple also deploys a strategy like this by developing the hardware, operating system, and software that are used by their devices.

The term *vertical integration* refers to a situation where one vendor controls many different components of the same supply chain. In Google's case, the chain is mostly software-based, whereas in Apple's case the chain is from assembly to use of a product. While these two companies in particular are often criticized for their vertical integration strategies, as it can also be a form of vendor lock-in, it provides measurable value for them in ensuring users are retained in their existing and future ecosystem.

Development of scientific software is increasingly seen as closer to conventional business models than was previously realized. Books like *Marketing for Scientists* by Marc Kuchner, social media outreach for scientific software projects, and a newfound focus on design and outreach have underscored the relationship between mindshare and developing strong scientific collaborations. The trend toward vertical integration, while perhaps not always recognized as such, has long been present in scientific software—although often termed feature creep, rather than a conscious strategy.

In my own day-to-day work, I see this, too. (And calling it *vertical integration* instead of *feature creep* makes me feel better about it when I'm the one doing it!) Particularly when prototyping a new feature, the barrier to entry to adding that new feature to an existing codebase is much smaller than the barrier of creating a new project. On top of this, the problem of dissemination is drastically reduced. As long as software packaging remains the challenge it is today, this will be a strong motivator for feature creep.

Along these lines, a few months ago my co-editor Konrad Hinsen posted a blog entry describing the changes he found in NumPy, a numerical Python library, over the years of its existence (see http://khinsen.wordpress.com/2014/09/12/the-state-of-numpy). Recently, NumPy dropped support for a particular, somewhat obsolete API from early in its history; this has resulted in broken code that needs to be updated. NumPy benefits from being a library, from being decoupled from the updates of packages that rely upon it. Contrast this with an integrated set of packages, where the dependencies and the underlying software were released in lockstep. In that case, either the necessary APIs would likely not be removed, or their removal would be accompanied by an upgrade to all dependent packages. Yet, if every application developer had chosen to write his/her own implementation of an array library, the NumPy/SciPy ecosystem would simply not exist—sharing data between libraries would be nearly impossible. If you've attempted to share data between libraries in C/C++ or Java, you may have experienced forms of this difficulty, as different libraries assume different conventions for arrays.

So Why Should We Vertically Integrate?

The Sage project (sagemath.org) ships a bundle of all of the dependent software packages in a single suite; this enables them to control the versions distributed, avoid breakages, and perhaps most importantly it essentially solves the problem of distributing a diverse set of packages to all major platforms. If they need to develop a custom version of something, they can ship a modified version of the package with their distribution. This strategy will help them control the user's experience, ensuring consistency. This particular example somewhat blurs the line between bundling and vertical integration.

 Published by the IEEE Computer Society

Jamie Zawinski famously said, "Every program attempts to expand until it can read mail." As the infrastructure of a program grows, it becomes very tempting to bolt on new features. In the life of an imaginary scientific program, someone might develop helper functions for arrays—why manually loop over elements, when this function can be overloaded and written only once? And while you're at it, why not add on some derivative functions, and then even routines to easily serialize these arrays? Each of these will be tuned to your exact use case, with a minimum of features, and then new developers to your project will have access to them as a given, as part of the core routines they can build on.

This lends itself to dividing the broad types of software into *applications* (mainly meant to be used by an end user), *libraries* (mainly meant to be linked into other applications and combined with other libraries to create an application), and what could broadly be called *platforms* or *environments*, within which applications and libraries can be assembled. The vertically integrated stack is well-suited to the realm of environments, like programming languages (Python, for instance, is known for coming "batteries included") and domain-specific dialects (Sage), where the ecosystem is somewhat controlled, somewhat centrally defined, and where inserting new components into that ecosystem can be moderately challenging.

Providing individuals in an "environment" with the ability to call upon a wide variety of standard, integrated, and deeply coupled components encourages them to invest in that environment, that ecosystem. This encourages them to continue to grow that software ecosystem.

And yet, libraries are often the worst choice to be vertically integrated—libraries by their very nature are meant to be components in broader ecosystems, to be the components out of which application developers compose their applications. Feature creep brings with it the danger of conflicts, and perhaps most challengingly, the risk of preferred solutions to non-essential functionality. As libraries begin to make decisions—for instance, a networking library deciding in what form bits should be stored on disk—that impacts other areas of the calling code's responsibilities, so that impedance mismatches and minor conflicts may creep in along with those features. This can have secondary effects, but it might also cause application developers to reconsider the heavyweight approach in favor of something considerably lighter. Often this will lead to application developers taking the vertical integration approach on their own!

So Why Shouldn't We Vertically Integrate?

In my mind, the biggest threat that comes with vertical integration—which has plagued Apple, Python, and Google—is that some components are just kind of low-quality, or worse, unreliable. If one component, upon which individuals have come to depend, is prone to failure, or fails to meet standards and needs, it can result in individuals getting frustrated and either loudly disparaging or even outright abandoning the ecosystem. As an example, the Python standard library—long held up as a shining example of the batteries-included approach Python took to programming—has in recent memory been criticized as stagnant, or as needing much faster turnaround times than are available with the Python core language release schedule. Finding that something exists, and the disappointment that comes from finding out it won't work for a particular use case, is discouraging.

Furthermore, particularly in scientific software, projects suffer from a dearth of resources. The creeping temptation to vertically integrate, to add on new components and continue to develop outward from a software ecosystem's core, runs the risk of spreading yourself further and further. The counterexample, of course, is when the new features end up injecting a considerable amount of energy into a community—witness the growth of IPython (now Jupyter) following the development and widespread deployment of the Web notebook that now widely drives adoption. The greater the number of subprojects, components, and elements that are "insourced" to a project, the deeper the commitment from contributors and developers required to continue the advancement of the project as a whole.

Is Scientific Software "Different"?

Two attributes of scientific software *producers* and *consumers* provide some distinctions from traditional business or consumer software. Scientific software producers and scientific software consumers are in general much closer in skillsets and interests than industrial software developers are with traditional consumers; this means that individuals presented with a poorly performing or otherwise rickety stack of software are more likely to look at it and say, "Well, I could do better, I don't need this." The buy-in cost is much different for a piece of scientific software than for, say, an operating system. This can work either for or against the notion of vertical integration; a well-designed, smooth stack of software is less likely to push people away, but the larger the stack, the more overwhelming and off-putting it can be.

This leads into the other looming difference: motivations and rewards for scientific software developers are centered around mindshare and attribution, and much less around revenue streams. Seeking mindshare requires both being of value and being deeply ingrained, but not so ingrained as to be invisible. And yet, when revenue is relevant—for instance, licensing software or services for scientific software to universities—it can create incentive for developers and discouragement for consumers to use the software.

In reading over this article, as I prepare to submit it, I realize that it's really an exploration of the choices available, more than a direct opinion piece. There's no easy answer, no quick discussion of the right way to develop software, particularly

scientific software, but there are many different viewpoints to consider. I have pursued vertical integration in some of my software projects—and occasionally, this was successful. In other projects, it furthered distraction and resulted in (to paraphrase Larry Page) less wood behind more arrows. But perhaps writing this will give me perspective to avoid such mistakes? Time will tell.

These are a few of my thoughts, but I invite readers to discuss them further, among their collaborations, or to send me your thoughts directly. What are the tradeoffs between providing a full stack, versus components? What are the drawbacks?

Matthew Turk is a research scientist at the National Center for Supercomputing Applications (NCSA) at the University of Illinois and a research assistant professor in the Astronomy department at the University of Illinois, Urbana-Champaign. His scientific research interests are focused on the way the first stars in the Universe formed, how they interacted with and transformed their environments, and how the fundamental mechanism by which stars form changed from that first generation to subsequent generations. His interests in computational science and discovery are focused on enabling deeper questions to be asked of data, and how to broaden the reach of software communities and collectives in science. Turk has a PhD in physics from Stanford University. Contact him at matthewturk@gmail.com.

This article originally appeared in Computing in Science & Engineering, *vol. 17, no. 1, 2015.*

IEEE computer society

ROCK STARS OF 3D PRINTING

Jesse Harrington Au
Chief Maker Advocate, Autodesk

Brian Gaff
Partner, McDermott Will & Emery, LLP

Paul Brody
VP & Global Industry Leader of Electronics, IBM

3D Printing Will Actually Change the World! Are You Ready?

Every company needs to prepare and implement 3D printing in order to remain relevant in their industry! No one can sit this phenomenon out!

Get ready at Rock Stars of 3D Printing, a one-day event featuring the experts, early adopters, and visionaries that are driving this revolution.

Develop Your 3D Printing Strategy! Ask Questions. Network with Experts. See Exhibits. Shift Your Paradigms!

Here's a list of other Rock Star speakers for Rock Stars of 3D Printing:

- Paul Brody, Vice President and Global Industry Leader of Electronics, IBM
- Brian David Johnson, Futurist and Director, Future Casting and Experience Research, Intel
- Cliff Waldman, Council Director and Senior Economist, Manufacturers Alliance for Productivity and Innovation

17 March 2015

The Fourth Street Summit Center
San Jose, CA

REGISTER NOW

Early Discount Pricing Now Available!

computer.org/3dprinting

IEEE

FedRAMP: History and Future Direction

THE FEDERAL RISK AND AUTHORIZATION MANAGEMENT PROGRAM (FEDRAMP), DEVELOPED BY THE US GENERAL SERVICES ADMINISTRATION (GSA) IN CONJUNCTION WITH THE US OFFICE OF MANAGEMENT AND BUDGET (OMB) AND THE FEDERAL CIO COUNCIL, WAS LAUNCHED 6 JUNE 2012. Figure 1 provides a timeline of FedRAMP-related events starting from the announcement of the initial working group through the two-year launch anniversary. FedRAMP is the US government program to apply the Federal Information Security Management Act (FISMA) to cloud computing. Initially, skeptics warned that the program wouldn't gain acceptance and would become another government IT casualty. Yet FedRAMP has been so successful that many governments in East Asia, Northern Europe, and the Americas are using it as a model for their own cloud security programs.

Cloud computing promises lower cost and the ability to quickly scale resources up or down as workloads demand, leading organizations in both the public and private sectors to seriously consider moving their applications and data to the cloud. Concern about cloud security has been the number one obstacle to adoption, particularly in the public sector.

FedRAMP provides a comprehensive set of cloud security requirements and an independent assessment program backed by the chief information officers (CIOs) of the Department of Defense (DoD), the Department of Homeland Security (DHS), and the GSA. Cloud service providers (CSPs) that implement the required security controls and meet independent assessment requirements can be authorized for use by the federal government. There's no shortage of CSPs jockeying for what has become the most coveted and prestigious qualifier of cloud security. So far, more than 50 CSPs have either been authorized, or are far enough into the process that the FedRAMP website lists them as "in process."

Standardizing the Authorization Process

Since its 2002 launch, FISMA has required that all systems hosting US government data be authorized prior to being put into production. The authorization process is extremely comprehensive, and until FedRAMP came along, system owners had to go through the entire authorization process for each agency using their system, even if the system was exactly the same from one agency to another. FedRAMP standardized the process such that authorizations can be performed once and reused by multiple agencies. It saves both government and private sector CSPs a lot of time and money and enables fast adoption of new systems and services. According to the FedRAMP program management office (PMO), Amazon estimates that its FedRAMP authorization saves approximately $250,000 per assessment. The FedRAMP PMO estimates that assessments cost the US government approximately $250,000. With the launch of FedRAMP, now CSPs are paying for the assessments instead of the U.S. government. The authorized cloud systems cover at least 160 known FISMA implementations across the

LAURA TAYLOR

Relevant Technologies
ltaylor@relevanttechnologies.com

 Published by the IEEE Computer Society

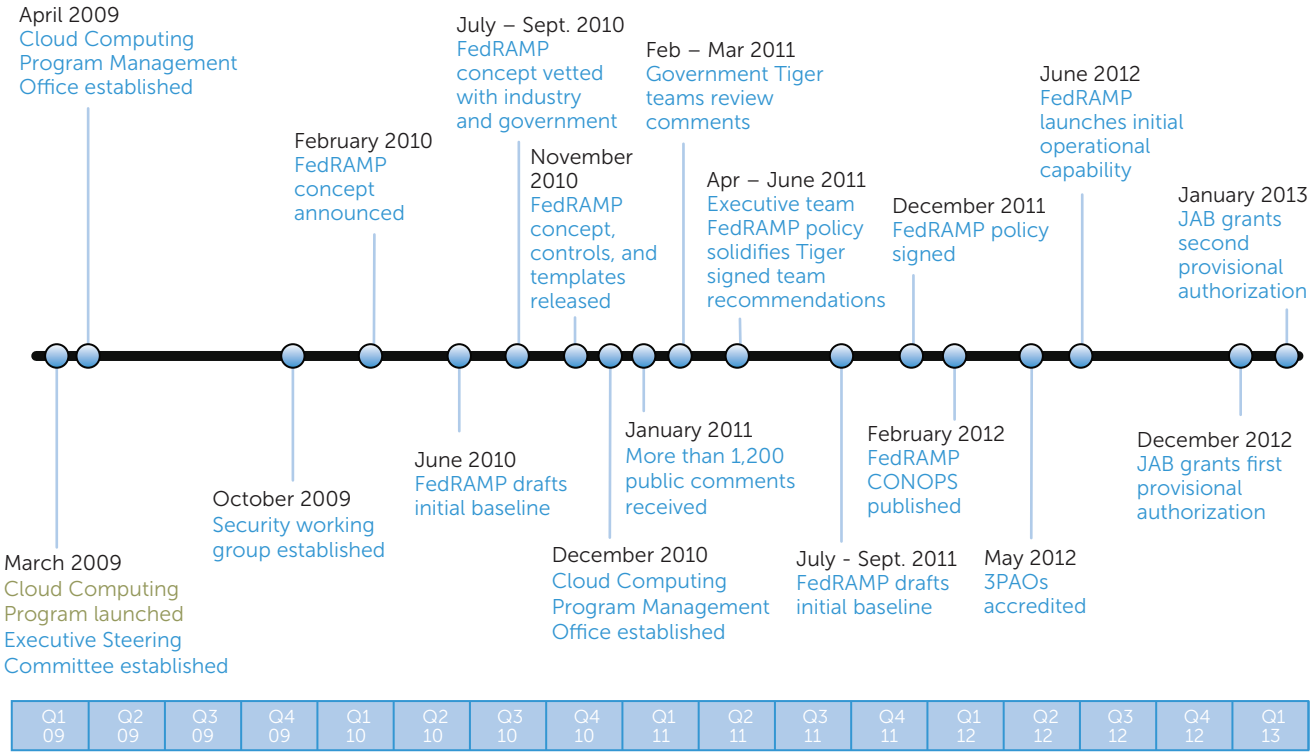


FIGURE 1. Federal cloud computing initiative and FedRAMP timeline. (Source: FedRAMP)

government, giving current FedRAMP cost savings a conservative estimate of $40 million dollars.

When he became the first federal CIO, Vivek Kundra championed cloud adoption as a way for agencies to save resources and improve service. However, without a way to secure the cloud and enable FISMA authorizations, cloud adoption would not come easily. Kundra launched the Federal Cloud Computing Working Group under the Federal CIO Council, a group of government CIOs that meets regularly to discuss government IT initiatives. At one of the council meetings, GSA CIO Casey Coleman volunteered GSA to take the lead in addressing federal agencies' adoption of cloud computing. Coleman in turn appointed her chief of staff, Katie Lewin, to manage the effort. In April 2009, GSA established the Cloud Computing PMO, and Lewin became the Federal Cloud Computing Initiative Director.

In addition to FedRAMP, Lewin was charged with heading up the Federal Data Center Consolidation Initiative (FDCCI). According to Lewin, "The FDCCI project was really the camel's nose under the tent for launching government cloud and ultimately FedRAMP." Lewin ensured that the brain trust at the National Institute of Standards and Technology (NIST) was involved with FedRAMP from the start.

The Federal Cloud Computing Working Group was initially chaired by Peter Mell. Mell was part of the NIST Information Technology Laboratory in the Computer Security Division and became involved in the working group after writing the technical definition of cloud computing adopted by the government cloud program.[1] In fall 2009, the group identified cloud authorization as the largest security hurdle to government cloud adoption. To address this, Mell conceived of the notion of government-wide authorization and worked out a formal process with his NIST colleagues (such as Ron Ross). He presented "A Notional Process on Security Assessment and Authorization for Cloud Computing Systems" to the working group and the cloud PMO. It was well received and, in early 2010, Lewin worked with Mell to present the idea to Kundra and then to the CIO council.

Forming a Cloud Policy

To pitch the idea, they needed a name. The acronym FedRAMP appeared on a paper plate next to Mell's sandwich

Table 1. FedRAMP preparation requirements.

Checklist	Description
1	You have the ability to process electronic discovery and litigation holds
2	You have the ability to clearly define and describe your system boundaries
3	You can identify customer responsibilities and what they must do to implement controls
4	System provides identification & 2-factor authentication for network access to privileged accounts
5	System provides identification & 2-factor authentication for network access to non-privileged accounts
6	System provides identification & 2-factor authentication for local access to privileged accounts
7	You can perform code analysis scans for code written in-house (non-COTS products)
8	You have boundary protections with logical and physical isolation of assets
9	You have the ability to remediate high risk issues within 30 days, medium risk within 90 days
10	You can provide an inventory and configuration build standards for all devices
11	System has safeguards to prevent unauthorized information transfer via shared resources
12	Cryptographic safeguards preserve confidentiality and integrity of data during transmission

Source: *Guide to Understanding FedRAMP*[5]

one day as he listed descriptive words for government-wide authorization programs. The logo (similar to the one used today) was a result of an internal security working group competition. The idea was adopted and the admittedly slow process of creating the first government-wide authorization process began.

In December 2010, Kundra published the 25 Point Implementation Plan to Reform Federal Information Technology Management.[2] The plan announced the cloud first policy, which stated, "When evaluating options for new IT deployments, OMB will require that agencies default to cloud-based solutions whenever a secure, reliable, cost-effective cloud option exists."[2] In February 2011, Kundra released the Federal Cloud Computing Strategy, which stated that government agencies must focus on managing services rather than assets.[3] In this paper, Kundra estimated that $20 billion of the then $80 billion in IT spending could be migrated to the cloud. Kundra forecast that by moving to the cloud, government agencies could improve server utilization by 60 to 70 percent and could increase responsiveness to urgent agency needs. The stage was set and government agencies would have to start migrating to cloud, like it or not (see Figure 1 for a chronology of events).

In August 2011, former Microsoft executive Steve Van Roeckel succeeded Kundra as federal CIO. Van Roeckel established FedRAMP via the "Security Authorization of Information Systems in Cloud Computing Environments" memorandum issued on 8 December 2011 (see https://cio.gov/wp-content/uploads/2012/09/fedrampmemo.pdf), which provided a cost-effective, risk-based approach for the adoption and use of cloud services. Under Lewin's leadership, the FedRAMP PMO ramped up quickly on resources when an OMB examiner transferred resources from the GSA Federal Acquisition Services (FAS) office to Lewin. Lewin hired Matthew Goodrich as deputy program manager. Lewin retired from the government in 2013, and Goodrich is the current acting FedRAMP director.

Although FedRAMP has attracted Amazon, Microsoft, HP, IBM, AT&T, and other big players, the first CSP to be authorized was Autonomic Resources, a government-only CSP headquartered in Research Triangle Park, North Carolina. Autonomic Resources predicted early that FedRAMP and DoD authorizations would be a boon to business and built a government cloud specifically for FedRAMP authorization. According to James Bowman, Autonomic Resources' government compliance director, "The ARC-P IaaS government community cloud solution was designed and built from the ground up to meet the stringent FedRAMP and DoD security control requirements. Our value lies in our cost savings, custom-built cloud services for government, and our high level of security and compliance."

Achieving FedRAMP Authorization

FedRAMP doesn't certify or authorize products of any kind. Rather, it aims to verify public and private cloud systems' security through FISMA. All US government clouds, private and public, must comply with FedRAMP. A system must already be built for its security to be verified. FedRAMP doesn't care what products you use to build your cloud as long as the system is secure, and as long as it meets the FedRAMP security control baseline. The Joint Authorization Board (JAB, which consists of the DoD, DHS, and GSA) selected the controls from NIST SP 800-53, *Security and Privacy Controls for Federal Information Systems and Organizations*.[4] The *Guide to Understanding FedRAMP*, V2.0, June 6, 2014 includes a preparation checklist (see Table 1).[5] If a CSP system can't at the minimum meet these requirements, it isn't a suitable candidate for FedRAMP.

Before FedRAMP, the authorization process inherently had many redundancies that duplicated authorization work from one agency to another. One agency didn't necessarily trust another agency's authorization process because it used different controls and security templates, and the independent assessment process differed from agency to agency. Even if an agency had authorized a cloud platform, each time a new agency wanted to use that platform, the CSP had to go through the authorization process all over again, as Figure 2a illustrates.

With the advent of FedRAMP, agencies now use the same security control baseline, the same security templates, and the same independent assessment process as illustrated in Figure 2b. The new process ensures consistency across all government agencies and instills a reciprocity of trust between agencies. Once a CSP has been authorized, any agency can leverage that authorization without repeating the process.

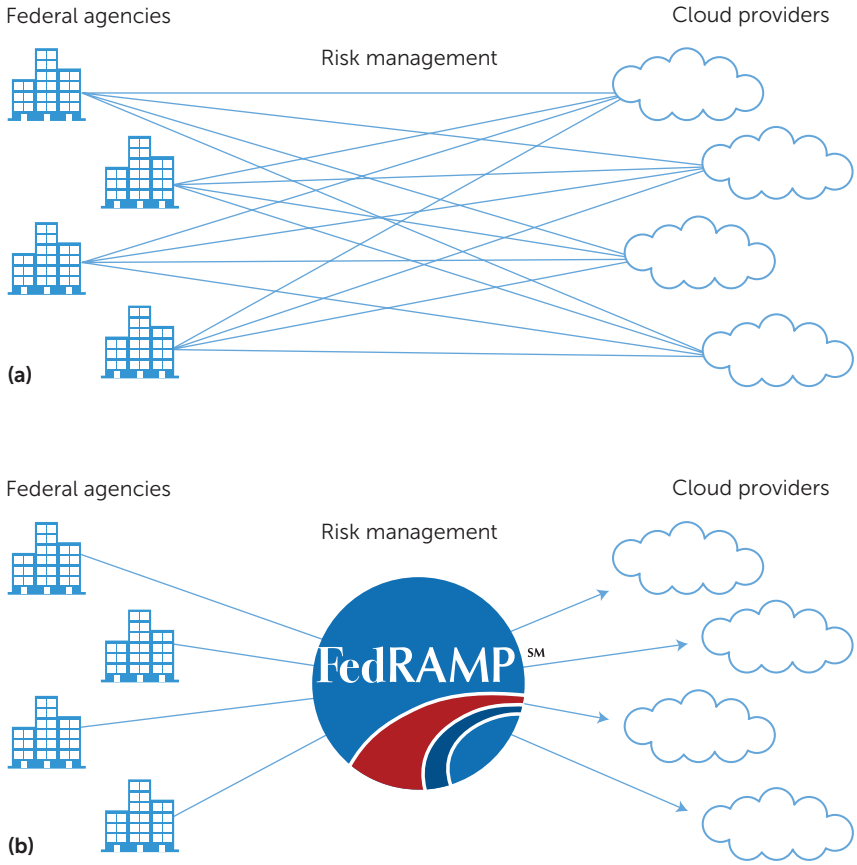


FIGURE 2. Authorization process for federal clouds: (a) old way and (b) new way. (Source: FedRAMP)

This new approach speeds up an agency's ability to roll out cloud services while reducing the cost of the authorization. The Department of Health and Human Services (USDA), the Department of Transportation (DOT), the Department of Agriculture (USDA), and the Department of Housing and Urban Development are at the forefront of cloud adoptions.

CSPs can use three different avenues to become authorized under FedRAMP. They can be authorized by the JAB or by an agency directly, or a CSP can self-submit a security package as a candidate for authorization. The FedRAMP website lists the three security package types. As of this writing, no CSP self-submitted packages are listed, although multiple CSPs are currently putting together packages in that category.

Once an agency decides to authorize a candidate package, the package moves to the "agency authorization" category on the FedRAMP website. The primary difference between an agency authorized package and a JAB authorized package is the level of review it undergoes. Agency-authorized packages are reviewed by one agency,

Increased level of review

Category	Description
CSP	CSP supplied, not yet reviewed (candidate for authorization)
Agency	Reviewed and authorized by agency
JAB	Reviewed by FedRAMS ISSO and JAB, authorized by JAB

FIGURE 3. Summary of FedRAMP packages.

whereas JAB-authorized packages are reviewed by the DHS, DoD, and GSA CIOs and their technical teams. The JAB's technical review teams consist of up to a dozen people from DoD, DHS, and GSA, all looking at the Security Assessment Report from different angles. Because of the number of people that review packages slated for JAB authorization, it can take considerably longer to get through the FedRAMP process if going through the JAB. Once a security package is listed in the FedRAMP repository, federal agencies can review it to determine if they want to use the system described in the package.[5] Figure 3 summarizes the three FedRAMP security package types described above

CSPs should not presume that their work is done after their system has been authorized. Continuous monitoring is required. According to Goodrich, "What we've seen at FedRAMP is that the hard part of security is people and processes, not the technology. The alignment of business processes like configuration management and patch management with vulnerability scanning is critical to a successful implementation of security on all systems." Authorized CSPs must perform monthly scans and send the scan results to their government authorization point of contact. High vulnerabilities must be mitigated within 30 days and moderate vulnerabilities must be mitigated within 90 days. Failure to mitigate vulnerabilities according to these requirements could lead to a CSP having its authorization suspended or revoked. FedRAMP's Continuous Monitoring Strategy Guide is available on the FedRAMP website.[6]

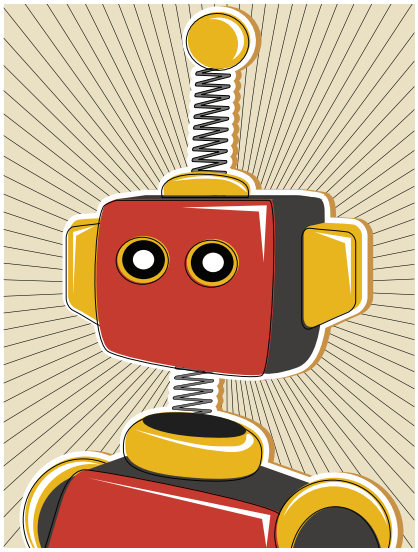
FEDRAMP WILL CONTINUE TO EVOLVE ITS PROGRAM AND PROCESSES OVER TIME. Check in at www.fedramp.gov for the latest updates.

References

1. P. Mell and T. Grance, *The NIST Definition of Cloud Computing*, National Inst. of Standards and Technology, NIST Special Publication 800-145, Sep. 2011, Nat'l Inst. of Standards and Technology; http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf.
2. V. Kundra, *25 Point Implementation Plan to Reform Federal Information Technology Management*, 9 Dec. 2010, The White House; https://www.dhs.gov/sites/default/files/publications/digital-strategy/25-point-implementation-plan-to-reform-federal-it.pdf.
3. V. Kundra, *Federal Cloud Computing Strategy*, 8 Feb. 2011, The White House; https://www.dhs.gov/sites/default/files/publications/digital-strategy/federal-cloud-computing-strategy.pdf.
4. Joint Task Force Transformation Initiative, *Security and Privacy Controls for Federal Information Systems and Organizations*, NIST Special Publication 800-53, rev. 4, Nat'l Inst. of Standards and Technology, Apr. 2013; http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf.
5. *Guide to Understanding FedRAMP*, Federal Risk and Authorization Management Program (FedRAMP) V2.0, 6 June 2014; http://cloud.cio.gov/document/guide-understanding-fedramp.
6. *Continuous Monitoring Strategy Guide*, Federal Risk and Authorization Management Program (FedRAMP) V2.0, 6 June 2014; http://cloud.cio.gov/document/continuous-monitoring-strategy-guide.

LAURA TAYLOR *is the founder of Relevant Technologies and the chair of the FISMA Center's Advisory Board. She specializes in security compliance and security audits of government agencies and financial institutions. Taylor has provided information security consulting services to some of the largest financial institutions in the world including the IRS, the US Treasury, the US government-wide accounting system, and various regional banks. She has also served as director of security research at TEC, chief information officer of Schafer Corporation, director of information security at Navisite, director of certification and accreditation at COACT, and director of security compliance at USfalcon.*

This article originally appeared in IEEE Cloud Computing, *vol. 1, no. 3. 2014.*



Turning Video Gamers into Software Developers

Jason W. Bay, ***GameIndustryCareerGuide.com***

Can attracting gamers help alleviate the tech talent shortage across the US? An industry professional offers some hope.

The United States faces a critical shortage in software development talent. Some estimates put the hiring gap in my home state of Washington alone at 20,000 unfilled technology jobs—and growing. At a recent event sponsored by the Technology Alliance, respected tech theorist Geoffrey Moore lobbed a passionate call to action: If we don't find ways to educate more middle-class kids into tech jobs, the very fabric of American democracy is in danger.

In creating and operating GameIndustryCareerGuide.com, my video game career website launched in 2013, I've learned many lessons pertinent to Moore's warning. But perhaps the most interesting and useful is that kids are deeply passionate about playing games, and their passion can readily be channeled into an interest in *making* games. Because making games largely involves computer science, I believe video games can serve as a sort of benign Trojan horse, winning young people over to focus on computer science careers. This is a massive opportunity we must take full advantage of, if we're to fill the hiring gap Moore describes.

DISCOVERING A NEED

The Game Industry Career Guide is now a resource that attracts more than 60 thousand page views monthly by gamer geeks around the world. But it started as a much smaller side project, meant only to fill a tiny niche that I discovered while working as a game developer in Kirkland, Washington.

Over my eight years as technical director for a video game studio, I spoke with hundreds of college students interested in careers making games. Each time I visited a campus, I found students hungry for information: how the industry works, how to get a job, how to succeed in what can be a frustratingly insular profession. Most of my talks morphed into Q&A free-for-alls, and afterwards a line of students would remain to ask more questions or to hand me their resumes, hoping to hobnob their way to the top of the hiring queue.

I was struck by the number of budding game makers who in no way considered themselves "software developers." Many didn't even realize they were teaching themselves computer science; they started as passionate game players, and only learned to write code so they could create their own games. Fortunately, many do discover that they enjoy the programming process in its own right, and shift their field of study to CS or to a video game degree program equivalent.

I enjoyed serving as a resource for these passionate students, and it was a good way to scout the best talent for my engineering department. But the growing popularity of post-secondary game development programs hinted at thousands of young people across the country hungry for information about game careers—each one an opportunity to turn a game player into a game maker. I started thinking about how I could reach a larger audience.

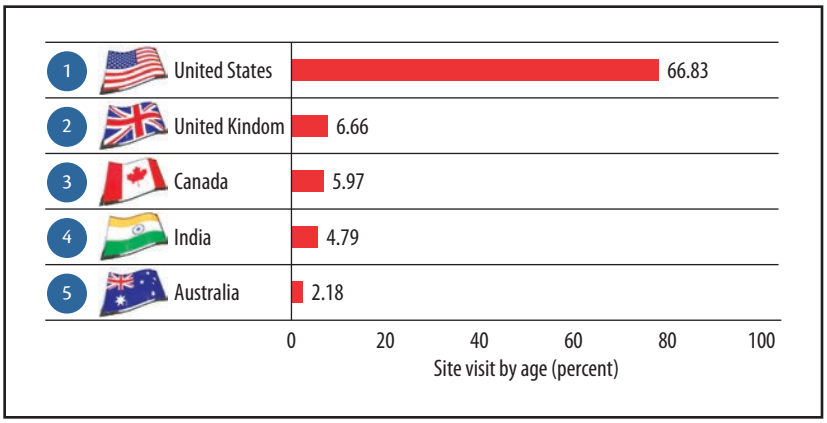


Figure 1. Sessions on GameIndustryCareerGuide.com for top five countries (percent). While most visitors to the site are in the US, the 33 percent traffic from other countries indicates interest in game development careers among English speakers throughout the world.

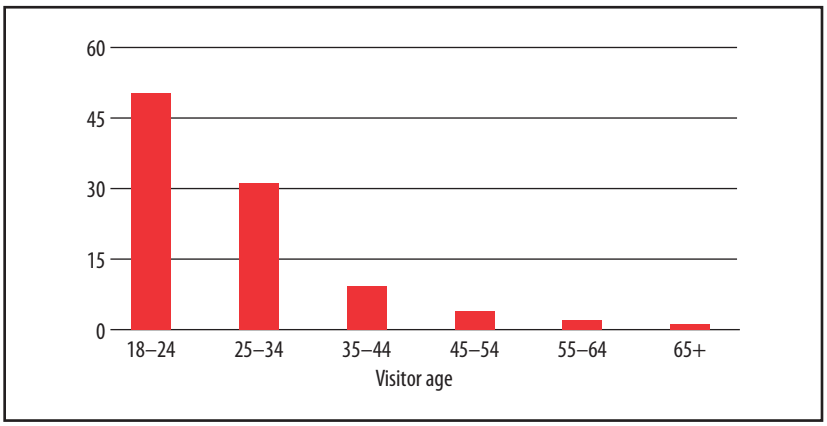


Figure 2. Sessions on the site by visitor age (percent). About half of all visitors to GameIndustryCareerGuide.com are under 25, indicating that high school- and college-age students have the strongest interest in game development careers.

GOING GLOBAL

The solution I devised was to create an online resource sharing my hiring-manager insights and experience with students everywhere. I began work on the Game Industry Career Guide, a Web resource providing information, advice, and inspiration to help game development students get a job after graduation.

After the site launched in May 2013 with just a handful of articles, traffic increased sharply and continued to grow. While I expected an interested audience looking to better understand the black box that is the video game industry, I certainly didn't anticipate such demand: in less than a year, the site began attracting readers internationally, as shown in Figure 1, garnering nearly 2,000 page views daily.

WHO WANTS TO MAKE GAMES?

I originally built the website to help graduating students succeed in their job searches, and so the early articles offered primarily job-related content on matters such as building a good resume, interviewing, and negotiating salary. But shortly after launch, I noticed that many incoming emails and blog comments were from high school students who loved games, and were turning to the Internet to investigate career options. Figure 2, which charts visitors to the site by age, suggests this interest.

Considering that more than 90 percent of kids aged 12 to 18 play video games almost daily, perhaps it's not surprising that they'd also be interested in making games for a living.

But despite the many careers available in the game industry, this high school contingent seemed familiar with only the two best-known: game designer and game tester. They didn't have a clue about the possibilities in programming, art, production, and other areas. It became obvious that I needed to provide a window into the full variety of career paths the game industry offers.

A NEW QUEST

That's why I launched the "Quest for Your Career" series of articles: to help kids understand the different jobs available, learn about each one, and start considering which might be a good path for them. Each article is built around an interview with an industry veteran with experience in a particular job who can offer role-tailored career insights. I drew from my network of game developer friends for the interviews, then edited their answers, formatted the articles, and emphasized key advice with pull quotes site visitors can easily share with friends via Twitter.

The series so far has proved a great success. With only 13 of 35 planned career-specific articles complete, it already accounts for a large percentage of the site's traffic. I expect it to become the site's central offering as I flesh out content for the remaining careers on my list.

HOBBY TO BUSINESS

When I set out creating the site, I hadn't intended it as a business. My goal was simply to get my thoughts and advice onto the Internet in

hopes of making life easier for budding game developers. Profit didn't cross my mind.

So when a company contacted me later that summer asking whether I would integrate their post-secondary school search software into my website—and get paid for doing so—I was skeptical. At that time, I was getting barely 1,000 page views per month, so I doubted my site could generate enough leads to be worth the effort of integrating their code. But since many of the visitors were graduating high school seniors beginning their search for a college or university, it seemed a good service to provide. I signed the agreement and integrated the school search into the site.

As the site's traffic grew, it became clear that this was indeed a service visitors valued: my leads gradually increased, and the revenue followed. By May 2014, just a year after I launched the site, I was receiving enough income from lead generation that I had to apply for a business license. The site has now become a viable side business, worth building upon for the future.

More important, I continue to get comments and emails from young people thanking me for providing the information and helping them start their game-making careers.

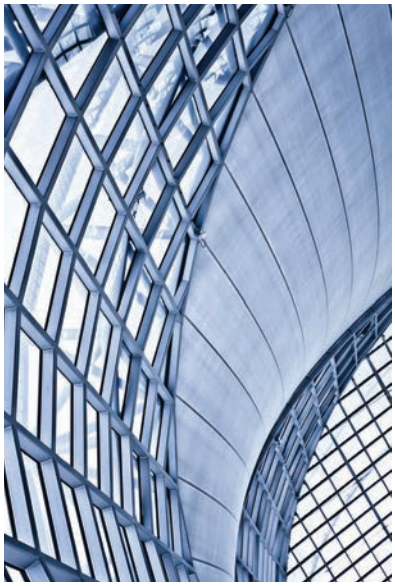
Despite dire warnings from mainstream media, perhaps games aren't rotting the minds of our youth after all. Instead, maybe they're one answer to our growing tech hiring gap. Maybe they're inspiring the next generation of software engineers, and all we need is to channel that enthusiasm and guide young gamers in the right direction. I hope the Game Industry Career Guide helps many take their first steps down that path.

Jason W. Bay spent 13 years in the video game industry as a designer, programmer, director, and studio head. He currently runs the GameIndustryCareerGuide.com website and is a senior product manager at www.Bonanza.com. Contact him at bayjason@hotmail.com.

This article originally appeared in Computer, *vol. 47, no. 10. 2014.*

Editor: Kelvin Sung, Computing and Software Systems, University of Washington Bothell; ksung@u.washingtonl.edu

EDITOR: Joseph Williams, Seattle Pacific University, josephwi@spu.edu



Microsoft CIO Talks Innovation, Career Growth, and Dogfooding

Joseph Williams, *Seattle Pacific University*

Jim DuBois' CIO role at Microsoft is unique in ways many tech professionals don't think about. Sure, he's responsible for applications, infrastructure, and solutions. Yes, he leads the company's information risk management and manages processes that keep the lights on for more than 127,000 employees in more than 110 countries. And of course he manages an IT organization with several thousand people and rather complex relationships with suppliers that are often key Microsoft customers and partners. All of this is important stuff.

Yet what makes his role extremely interesting is that his team deploys the pre-release versions of Microsoft's products directly into the company's production environment. His team and Microsoft's 100,000 employees provide the proving ground for products like Microsoft Office and Azure before they are released, which means the company is running on software that sometimes isn't quite ready for prime time. Nevertheless, DuBois' team makes it work while providing direct feedback to the Microsoft product groups so they can improve and enhance the products before their commercial release.

A 20-year Microsoft veteran, DuBois became CIO on 1 Jan. 2014, after serving as the interim CIO for seven months. A graduate in computer science from the University of Washington, DuBois spent seven years at Accenture before following his mentor to Microsoft. Although he relished the consultant's life, after the birth of his second daughter, DuBois was looking for a more stable work life, and Microsoft offered some interesting challenges. Although Microsoft moved DuBois around with tours in Tokyo and London, he was able to take his family with him. He wasn't intentional about getting international experience so he could become a CIO, but in retrospect, it provided him with great perspective about local and regional organizations that have helped inform his views on IT management.

What was it like for you, personally, when you were first promoted to CIO?

It's been a sea change, not only on the personal side but also at the company and industry level. I've never learned so much in my career—first and foremost, learning to be uncomfortable.

The biggest impact was when I was asked to be interim CIO. Kevin Turner, our chief operating officer and who I report to now, was clear that it was an interim role while they looked for a permanent candidate. During this time we had major changes at the company: a new strategy, a major acquisition, and a new employee performance review program, not to mention the announcement of a retiring CEO. All these changes were exhilarating to me in the role of CIO, because IT was the key enabler to the first three: the devices and service strategy, the Nokia acquisition, and the new employee review model. So change was all around me.

I asked the CIO leadership team to allow me to play the role of CIO, as we could not afford to lose momentum, and we couldn't wait until a new leader was appointed. I really dug into the role while I prepared myself for bringing a new leader

 Published by the IEEE Computer Society

up to speed. The interim CIO role meant I had to partially transition away from day-to-day operational management and project execution oversight to a role focused more on strategic planning, partnerships with a broader set of executives, and IT staff management.

At home, the interim CIO role meant my wife and I had to adjust to new time demands while supporting priorities at home. My wife is in the midst of getting her doctorate in psychology, so our personal life was already busy.

After seven months, Kevin told me that I'd changed his view of what I could do and he'd not found anyone else who could do the job better. I accepted the permanent CIO role in January. The interim role really helped me appreciate the honor of being asked to take the role and to lead the people within Microsoft IT.

What is the biggest personal challenge of your job as CIO?
It's true that the IT team has big challenges and opportunities ahead with the Nokia integration into Microsoft, and many other strategic projects. But as I grow into the role of CIO, I know leadership of people and staff management are priorities that can't be minimized or put on the back burner. In my brief time as CIO, the number of IT employees has grown by 30 percent. So, the biggest challenge for me is to develop an IT culture that enables the Microsoft transformation in a mobile-first, cloud-first world, and one that reflects the cultural attributes of a One Microsoft approach.

What is it like being the CIO for a company full of brilliant technologists, many of whom could be or have been CIOs in their own right?
It's a challenge and an opportunity—my peers in the high-tech industry can empathize.

Jim DuBois: Interesting Tidbits

Best book recently read: *How Will You Measure Your Life?*, by Clayton Christensen (HarperBusiness, 2010)
Nonviolent Communication: A Language of Life, by Marshall Rosenberg (Puddledancer Press, 2003)—recommended by Microsoft CEO Satya Nadella
How Stella Saved the Farm, by Vijay Govindarajan (St. Martin's Press, 2013), which uses a parable to teach lessons about making innovation happen

Favorite industry publication: I don't read any one publication regularly. And depending on where I am in the world, I find myself reading something different. I spend most of my time reading whatever my friends on the Internet like and recommend.

Last time you actually coded or configured: Outside of work, it was last year. At work, they haven't let me touch anything for years now.

What mobile phone do you carry?: Nokia Lumia 1020. I like the personalization, the great camera, and Cortana.

What other personal technology do you use?: Xbox One—my favorite game is Titanfall.

I expect Microsoft employees to work with pre-release versions of Microsoft products and services.

I love being surrounded by smart, diverse, and motivated people. They're more technical than the general population and expect that we provide IT services accordingly. It allows me to play a bigger role than I might in a typical company, because some of our IT services are run by product engineering groups, and there are many employees who self-manage their devices. It means that Microsoft IT needs to trust employees to do the right thing, and tolerate more risk than IT staffs in most other industries. It also means that I have higher expectations of Microsoft employees to educate themselves on how to use devices wisely and how to treat different types of data. And that means that I expect Microsoft employees to work with pre-release versions of the Microsoft products and services so they can help us provide feedback on their experiences. The IT staff manages this process, which is called "dogfooding" internally, because we eat our own dog food before it's available to customers.

Where does "innovation" play in your role? Are you a catalyst?
Microsoft IT innovates on devices and services. It's one of the three core IT strategies. This strategy appears in several ways. First, we continuously improve the service and support experience for all Microsoft external customer segments.

Second, we want to inspire customers with our innovations

on devices and services within Microsoft, such as our hybrid cloud strategy or how we enable and manage a bring-your-own-device (BYOD) work policy. We share those experiences via online case studies and via 6,000 customer meetings this year.

Finally, we work closely with the product engineering teams to design, run, test, and provide feedback on Microsoft's commercial products. Our work of being the first enterprise customer of Microsoft's devices and services ensures that Microsoft's customers have a better experience with Microsoft's products.

Jerry Seinfeld had Newman. Sherlock Holmes had Professor Moriarity. Who is your nemesis?
Without naming names, I'd say time. Now that I'm CIO, it's a challenge to find the time for all the priorities in life. In my old role, I could prioritize meetings with all the product engineering and R&D teams to see what's coming and how these products could be used to run the business. I get to attend fewer of those meetings now and have delegated attendance to others.

I also say time because one of the roles of IT is to make the company faster. From getting products to market faster to helping customers find the answer to their support issue faster, to getting prospective employees into the interview loop faster and helping sales teams qualify leads faster. Speed is an imperative for IT employees.

What is the one thing you need to delegate that you haven't yet?
Technical support for my friends and family. Just the other day I was helping my dad with his online access and authentication into MSN. I suspect everyone in IT can relate.

But seriously, shedding my favorite and comfortable responsibilities from my old role has been a challenge with the transition to CIO. My assistant, business manager, and HR partner remind me of this regularly.

How do you personally keep up with technology in general?
Keen interest and osmosis. I have a keen interest in the ability of technology to augment life by making it better, simpler, and more enjoyable. I'm lucky to be surrounded by some of the smartest technology people in the world, which makes it fun to come to work every day. A recent example is from Juha Turkki, the former CIO of Nokia and who's now on my staff. He's teaching me about Nokia devices, services, and apps and how he uses them.

How do you personally keep up with the technical details of Microsoft's vast product line?
The real answer is that I don't keep up with everything, but I'm inquisitive and love to learn and truly want to know the impact and outcomes of IT engagements with the business. One of these engagements is how we "dogfood" products inside the company to drive business initiatives forward.

A few months ago, we put into production a feature of Microsoft Azure called ExpressRoute. This feature is going to allow IT to securely move hundreds of applications to the cloud per month, giving us more flexibility. As I think about retiring thousands of end-of-life servers over the next few years, the move to the cloud reduces costs and increases business agility. That's a technical feature any CIO would want to know about.

What part of your job as CIO gives you the most satisfaction?
I love being in the middle of everything we are trying to do as a company. The role of technology is more important than ever in any company's success. I love playing the role of customer with our product teams, and exchanging ideas and best practices with industry CIOs.

What are the most valuable lessons you've learned in your career?
Helping people understand they are part of something big and impactful is rewarding. Making really big impacts requires change. A lot of people don't like to change. Explaining change is hard, because everyone hears the message through their own filters, based on their own life experiences. I'm still learning how to do this better, but I know it takes leadership, role modeling, persistence, and finding the right, influential people open to change early.

What is your advice to aspiring IT professionals who wish to move up to the ladder to the C-Suite?
There are many paths up the ladder. But as I reflect on my journey from IT pro to CIO, some items come to mind. First, I've always had a plan for what I want to do next, but it has almost never turned out the way I planned because opportunities surfaced that didn't

One of the roles of IT is to make the company faster. From getting products to market faster [to] helping sales teams qualify leads faster.... Speed is an imperative for IT employees.

align with my plan. Take advantage of opportunities to gain new experiences rather than always looking for the next step up on the ladder. Getting experience in many areas has prepared me more for advancement than anything else.

Second, know the business you support and help solve business problems. Speak the language of the business and be an advisor on how technology can deliver solutions. I'm reminded of some coaching I received along the way—that the skills that got me to this new role will not be the same skills that will make me succeed in the new role. There's always room for learning and growth.

Tell me about a typical day in your life.

I'm not a morning person and, if possible, I try to take my son to school. So if I can help it, my work day doesn't start before 9:00 a.m. I try to spend at least 10 percent of my time with customers, then divide up the rest between running IT and working with our product teams. I try to set aside an hour a day for thinking and strategy, but right now I don't stick to that rule most days. I'll answer emails in between meetings or lulls in meetings, and post to Yammer (internal enterprise social site) a couple times during the week. Sometimes I have lunch, although usually lunch is with a meeting. I tend to work later at night after my family goes bed. I like this time to read, catch up on some of my mail, post to Yammer, and prepare for the next day. I've always done some work at night. It's the balance I chose as I'm up late anyway. As for the weekends, I try to limit that as best as possible, especially if it's a sunny Seattle day.

How often do you see the CEO/COO/CFO?

I'm fortunate that I used to work with Satya Nadella in my old role and when he ran Microsoft's enterprise and cloud business. As a new CIO with a new CEO, every week Satya expects me to let him know what I have learned as a customer of our devices and services. I have a good connection with our CFO, Amy Hood, so we can accomplish most of what we need to via email as she helps me with corporate governance. I report to our COO, Kevin Turner, and interact with him regularly as well.

What is the best part of your day?

It's different every day. There are times where I know we are getting stuff done that is needed. There are better times when I get to make a decision, influence a direction for the company, help someone else be successful or teach them something new so they can make a bigger impact—where I really feel like I earned my keep. Those are the times I value most.

Do you take vacations and, if you do, are you plugged in?

My family loves to travel. We try to get away somewhere new every year. I do like to stick around in the Seattle area in the summer and take three-day weekends rather than long vacations. On the family adventures, I completely unplug from work. Other than that, I do stay connected. Technology is part of my life and helps me make the most of life experiences.

How long do you think you can keep up this pace?

I love it, and I'm all in. It's been a little over a year now as CIO, including the interim role. I can't think of anything I'd rather be doing.

Joseph Williams *is the Dean of the School of Business and Economics at Seattle Pacific University. Contact him at josephwi@spu.edu.*

Take advantage of opportunities to gain new experiences rather than always looking for the next step up on the ladder.

This article originally appeared in IT Pro, *vol. 16, no. vol. 4. 2014.*


Notes from the Community

Editors: Mary Baker ■ HP Labs ■ mary.baker@hp.com
Justin Manweiler ■ IBM T. J. Watson ■ jmanweiler@us.ibm.com

From 3D Printing to Spy Cats

Mary Baker, *HP Labs*
Justin Manweiler, *IBM T.J. Watson*

Welcome to the latest column of Notes from the Community. We start by thanking Jason Hong for his past authorship of this column and introducing Justin Manweiler as the new department coeditor.

Members of our Reddit community submitted many fascinating items this quarter. This issue's topics include the various ways we try to translate between virtual and physical reality, technologies that increase our enjoyment of playing and watching sports, new techniques for spying (with both good and bad results), and, of course, wearables—even wearables for feline espionage!

CREATING NEW REALITIES

Our readers continue to enjoy discussing new technologies for transforming physical reality into digital reality, and back again. Instead of ordering out for pizza, we'll soon be able to print a restaurant and also create its menu of tasty digital dishes.

3D Printing

A building contractor named Andrey Rudenko created a printer that translates CAD designs into concreate reality.[1] Using technology and software from the RepRap 3D printing project (http://reprap.org/), Rudenko's device puts out layers of sand mixed with cement with a high enough viscosity that walls can even include decorative elements (see Figure 1). The Minnesotan has demonstrated his printer by creating a single-story castle playhouse in his backyard, which took three months to complete. Next up will be a full-sized, two-story house—which presumably won't be printed in his own backyard.

Andrey Rudenko has demonstrated his printer by creating a single story castle playhouse in his backyard.

Virtual Lego Reality

The Lego Company is giving kids a way to bring together physical Legos and a virtual world. With Lego Fusion, players start out using the famous plastic bricks to build physical objects on a special platform equipped with a scannable barcode.[2] The kids then use a smartphone to run the Fusion app and snap a picture of their accomplishment. Their physical creations magically appear in a virtual world that players can share with their friends. As new needs are discovered virtually, kids can create and add new physical elements. We assume this approach requires less space and fewer emergency room visits than a concrete backyard castle, although it also provides fewer opportunities for exercise and spray-painting on walls.

Transforming the Virtual to the Physical

Moving in the other direction, the inFORM Dynamic Shape Display renders 3D virtual content in physical form. Each (rather large, coarse-grained) "pixel" on the physical grid can be raised or lowered to reflect the varying depth of the virtual surface. By changing the height of these pixels in real time, inFORM even allows interaction with objects in the real world through virtual means (see Figure 2). Watch a video of the display in action at http://tangible.media.mit.edu/project/inform.

Digital Lollipops

We've made good progress translating video, audio, and now 3D shapes between the virtual and physical, but what about taste? Past efforts to communicate taste digitally usually involved remotely controlled release of combinations of chemicals. That approach requires transporting and

JOIN OUR SUBREDDIT

This column offers a summary of interesting news and research in pervasive and mobile computing, with content drawn from submissions to a shared community on the social news site Reddit, at www.reddit.com/r/pervasivecomputing. We encourage you to join our subreddit and spread the news of this site to others, so that together we can build a sustainable online community for all aspects of pervasive and ubiquitous computing. —*Mary Baker and Justin Manweiler*

 Published by the IEEE Computer Society

storing the chemicals, and many people aren't comfortable with ingesting the results.

Instead, researchers at the National University of Singapore use a combination of noninvasive electrical and thermal stimulation on the tongue to produce taste sensations such as salty, sour, or bitter.[3] Goals of the work include entertainment applications, including "taste TV," where audiences can experience flavors along with the program's characters, and medical applications, including digital lollipops that allow diabetics and others with restricted diets to enjoy the sensation of foods they must avoid.

Figure 1. The ultimate playhouse—a castle playhouse created via 3D printing. (Source: Andrey Rudenko; used with permission.)

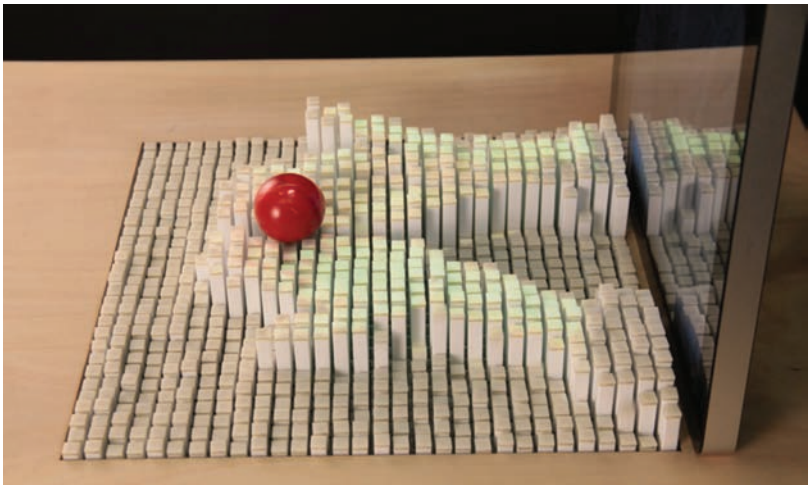


Figure 2. Physical interaction with the inFORM display. (Source: inFORM, http://tangible.media.mit.edu/project/inform; used with permission.)

EXTREME SPORTS (FOR SPECTATORS)

Sports have long motivated the invention of new technologies, and this quarter we find basketball and baseball driving efforts in extreme displays and camera work.

An LED Basketball Court

Nike and AKQA have collaborated to create an interactive LED basketball court floor.[4] The floor provides astonishing special effects for the audience, but it can also provide training help for the players, as Kobe Bryant demonstrated on his recent tour of China, where the court was unveiled. View the fancy footwork at www.youtube.com/watch?v=u2YhDQtncK8.

Action from any Angle

The YES Network is using Replay Technologies' FreeD video-production tool at Yankee Stadium. The current system, housed in the TV compound, uses a server connected to a couple of four-megapixel cameras located around the first-base side of the stadium. They combine these feeds to create 3D data that lets the operator move a virtual camera through a 3D space to examine a play from different angles. Sports Video Group reports that as of 3 June 2013, the system was mostly capturing plays at home plate, where you can swoop around the batter and see exactly how he moves and makes contact with the ball.[5] Watch the bat hit the ball at www.youtube.com/watch?v=iz9l7IlLtrQ&list=UU5fniymL41G6OiRRaQM9izg.

UBIQUITOUS SPYING

Where there are sensors, there can be spying, and this quarter, readers contributed links to some particularly controversial examples.

Tracking Down your Phone's Thief

There are several stories about victims of smartphone theft tracking down thieves through "find my phone" applications. While potentially dangerous, there's something satisfying about using the device itself to provide enough information to find its thief. The Reddit community was particularly fond of the story of Sarah Maguire, who retrieved her phone and her friend's phone from a man who had stolen them both the

NOTES FROM THE COMMUNITY

Figure 3. Rear-end wearables—a wearable chair that straps to your derriere. (Source: Noonee, www.noonee.ch; used with permission.)

Figure 4. Coco the cat has been fitted with a wearable spy collar. (Source: Gene Bransfield; used with permission.)

previous evening.[6] Law enforcement officials remind us that such adventures might not end well, and perhaps the phone isn't worth the possible repercussions of recovering it. This potential for violent confrontations is covered by a darker story from the New York Times that describes how GPS trackers in pharmaceuticals led police to the thief, who was then shot by police when he pointed a gun at them.[7]

Spying Runs in the Family

Many parents believe there are good reasons to spy on their teenagers, especially when they get behind the wheel of a car. These parents fear that smartphone apps that sense driving and turn off the driver's texting capabilities are not powerful enough. Esurance has come up with a more comprehensive solution, as reported by The Atlantic Citylab.[8]

The DriveSafe device fits into the diagnotstic port of gas-fueled cars made in the last two decades. It lets parents customize what teens can and can't do with their phones while driving. For instance, some parents might turn off texting and tweeting but still allow navigation apps to work. They might turn off the ability to place and receive calls entirely (other than 911 calls, which can't be disallowed), or they might allow only incoming calls from family members. The device also spies on the driver's behavior and reports it to parents. After a drive, parents can receive a message describing all the possibly unsafe behaviors the driver exhibited related to velocity, acceleration, braking, destinations, and so forth. DriveSafe will be free for all Esurance customers. Teenage joyriding might officially be a thing of the past.

AND, OF COURSE, WEARABLES

Conveniently timed for this special issue, readers contributed many links about various aspects of wearable computing, including some intended for rarely featured applications and body parts.

Interviews and Overviews

Reddit users posted many wearable-related items that aren't just descriptions of new gadgets. They posted a summary of experience by Kai Kunze, who reports on his grandparents' use of Google Glass.[9] (Kunze also coauthored an article in this issue with Paul Lukowicz, "Sensor Placement Variations in Wearable Activity Recognition."). In addition, there was a pointer to *Verge*'s interview with wearables pioneer Sandy Pentland,[10] as well as a link to an overview of many different kinds of wearables by

IEEE Spectrum.[11] Finally, there was a request for pervasive datasets to use in research—user "rrawasi" posted a request for datasets that include not just location information but also subjective input about users' moods (see www.reddit.com/r/pervasive-computing/comments/2c4bbh/looking_for_a_pervasive_dataset). It's good to see our subreddit getting this diversity of use.

Gadets!

This is not to say that readers neglected to report on new gadgetry. An *Engadget* article describes Docomo's attempt to create a more independent wearable by moving your phone's SIM card into a wrist device.[12] The article shows a picture of Acer's Leap band for motivation, as the Docomo prototype is currently the size of a Wi-Fi hotspot—which would make for an awkwardly large wrist wearable.

Also, Nod is developing a gesture-sensing ring to allow you to control everything around you that you could conceivably control with gestures, such as typing on virtual keyboards through swiping gestures, turning up virtual thermostats, using a thumb-reachable button on the ring to control your music, and playing games. In the video (see www.hellonod.com), gestures that require precision or low latency (such as keyboard entry and game playing) seem a little iffy. Maybe further development will improve matters.

And because not all wearables need to be tiny computers that strap to wrists, ears, fingers, faces, and feet, the subreddit also includes a link to *Engadget*'s story on noonee's wearable chair for factory workers, which provides power-efficient mechatronic posture support strapped onto one's legs and derriere (see Figure 3).[13]

Wearables for Spy Cats

Combing several of the topics discussed here, along with the Internet's fondness for cats, we have a link to *Wired*'s article on "How to Use Your Cat to Hack Your Neighbor's Wi-Fi."[14] The inventor, Gene Bransfield, fitted out his grandma-in-law's cat Coco with a WarKitteh collar that includes specialized firmware on a Spark Core chip, a Wi-Fi card, a tiny GPS module, and a battery (see Figure 4). As the kitty roamed the neighborhood, the system-on-a-collar detected many Wi-Fi hotspots and revealed that a third of them were either open or using broken WEP encryption.

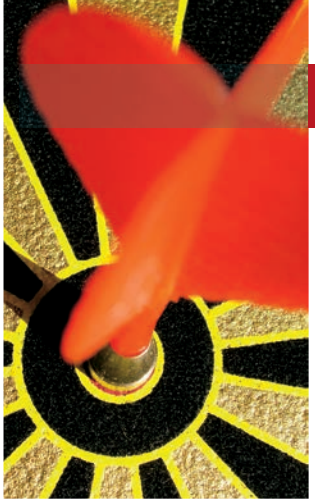
REFERENCES

1. M.-A. Ruson, "Man 3D-Prints Castle In Back Garden Using Concrete Printer He Invented," *Int'l Business Times*, 31 July 2014; www.ibtimes.co.uk/man-3d-prints-castle-back-garden-using-concrete-printer-he-invented-1459223.
2. D. Terdiman, "Lego Jumps from Physical to Virtual World, and Back Again," *CNET*, 19 June 2014; www.cnet.com/news/lego-jumps-from-physical-to-virtual-world-and-back-again.
3. "Yummy Nigella—How We Could Soon Sample TV Chefs' Food from our TVs," *Business Newsletter*, 22 Nov. 2013; www.independent.ie/business/technology/yummy-nigella-how-we-could-soon-sample-tv-chefs-food-from-our-tvs-29775482.html.
4. T. Brooke, "Nike's 'House of Mamba' LED Basketball Court Helps You Train Like Kobe Bryant," *Bleacher Report*, 16 Aug. 2014; http://bleacherreport.com/articles/2165406-nikes-house-of-mamba-led-basketball-court-helps-you-train-like-kobe-bryant.
5. K. Kerschbaumer, "Replay Technologies 'FreeD' Gives YES Network a New Perspective for Yankee Games," Sports Video Group, 3 June 2013; http://sportsvideo.org/main/blog/2013/06/replay-technologies-freed-gives-yes-network-a-new-perspective-for-yankee-games.
6. I. Lovett, "When Hitting 'Find My iPhone' Takes You to a Thief's Doorstep," *The New York Times*, 3 May 2014; www.nytimes.com/2014/05/04/us/when-hitting-find-my-iphone-takes-you-to-a-thiefs-doorstep.html.
7. J. Goldstein and J. Schwirtz, "Robbery Suspect Tracked by GPS and Killed," *The New York Times*, 17 May 2014; www.nytimes.com/2014/05/17/nyregion/robbery-suspect-tracked-by-gps-and-killed.html.
8. M. Riggs, "Will This Tiny Tool Stop Teens from Texting While Driving?" City Lab, 20 Dec. 2013; www.citylab.com/tech/2013/12/will-tiny-tool-stop-teens-texting-while-driving/7933.
9. K. Kunze, "Google Glass for Older Adults," blog, 24 Aug. 2104; http://kaikunze.de/posts/google-glass-for-eldery.
10. M. Konnikova, "Meet the Godfather of Wearables," *The Verge*, 6 May 2014; www.theverge.com/2014/5/6/5661318/the-wizard-alex-pentland-father-of-the-wearable-computer.
11. A. Bleicher, "Wearable Computers Will Transform Language." *IEEE Spectrum*, 28 May 2014; http://spectrum.ieee.org/consumer-electronics/portable-devices/wearable-computers-will-transform-language.
12. M. Smith, "Japanese Carrier Docomo Wants to Move your Phone's SIM Card into a Wearable," *Engadet*, 10 June 2014; www.engadget.com/2014/06/10/portable-sim-wearable-docomo.
13. C. Velazco, "Swiss Startup Creates a Wearable Chair So You Can Rest Anywhere," *Engadget*, 21 Aug. 2014; www.engadget.com/2014/08/21/chairless-chair.
14. A. Greenberg, "How to Use Your Cat to Hack Your Neighbor's Wi-Fi," *Wired*, 8 Aug. 2014; www.wired.com/2014/08/how-to-use-your-cat-to-hack-your-neighbors-wi-fi.

Mary Baker is a senior research scientist at HP Labs. Contact her at mary.baker@hp.com.

Justin Manweiler is a researcher at IBM. Contact him at jmanweiler@us.ibm.com.

This article originally appeared in IEEE Pervasive Computing, *vol. 13, no. 4, 2014.*



Editor: **Daniel Zeng**, University of Arizona and Chinese Academy of Sciences, zengdaniel@gmail.com

Man versus Machine or Man + Machine?

Mary (Missy) Cummings, *Duke University and MIT*

In developing any complex system that involves the integration of human decision making and an automated system, the question often arises as to where, when, and how much humans and automation should be in the decision-making loop. Allocating roles and functions between the human and computer is critical in defining efficient and effective system architectures. However, despite the recognition of this problem more than 60 years ago, in this case by NASA (see Figure 1), little progress has been made in balancing role and function allocation across humans and computers.

The problem of human-automation role allocation isn't an academic exercise or limited to a few highly specialized domains such as NASA. The rise of drones (or unmanned aerial vehicles) and the problems with remote human supervision are an extension of well-documented human-automation interaction problems in fly-by-wire systems in commercial aviation. Mining industries increasingly use automation to augment and in some cases outright replace humans, and robots that require human interaction are on the battlefield and in surgical settings. While these applications might seem far from everyday life, Google's recent announcement to introduce driverless cars to the mass market in 2017 and the race to develop in-home robots will make the human-automation allocation issue and associated computing demands ubiquitous.

The predominant engineering viewpoint across these systems is to automate as much as possible, and minimize the amount of human interaction. Indeed, many controls engineers see the human as a mere disturbance in the system that can and should be designed out. Others may begrudgingly recognize that humans must play a role in such systems, either for regulatory requirements or low probability event intervention (such as problems in nuclear reactors).

But how do we know what's the right balance between humans and computers in these complex systems? Engineers and computer scientists often seek clear design criteria, preferably quantitative and directive. Most engineers and computer scientists have little to no training in human interaction with complex systems and don't know how to address the inherent variability that accompanies all human performance. Thus, they desire a set of rules and criteria that reduce the ambiguity in the design space, which for them typically means reducing the role of humans or at least constraining human behavior.

A Brief Historical Perspective

In 1951, a National Research Council committee attempted to characterize human-computer interaction (then called human-machine interaction) prior to developing a national air traffic control system.[1] The result was a set of heuristics about the relative strengths and limitations of humans and computers (see Table 1), sometimes referred to as "men are better at" and what "machines are better at" (MABA-MABA).

The heuristic role allocation approach, exemplified in Table 1, has been criticized as attempting to determine points of substitution—because, for example, such approaches provide engineers with justification (possibly erroneously) for how to replace the human with automation.[2] For traditional engineers with no training in human-automation interaction, this is exactly what they're trained to do—reduce disturbances and variability in a system and make it more predictable. Indeed, they're trying to "capitalize on the strengths [of automation] while eliminating or compensating for the weaknesses,"[2] and this is an important piece of ethnographic information critical for understanding why traditional engineers and computer scientists are so attracted by such representations.

 Published by the IEEE Computer Society

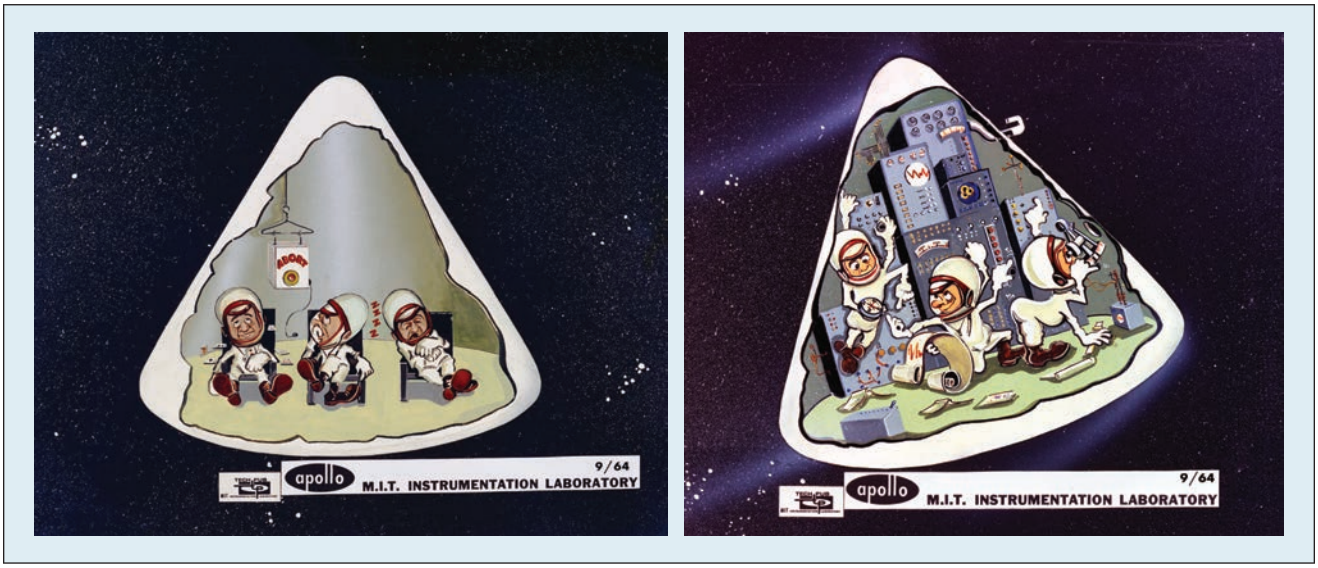


Figure 1. The role allocation conundrum for the Apollo missions. (Photos provided courtesy of The Charles Stark Draper Laboratory, Inc.)

Table 1. Fitts' list,[1] which characterizes human-computer interaction.

Attribute	Machine	Human
Speed	Superior	Comparatively slow
Power Output	Superior in level in consistency	Comparatively weak
Consistency	Ideal for consistent, repetitive action	Unreliable learning and fatigue are factors
Information capacity	Multichannel	Primarily single channel
Memory	Ideal for literal reproduction, access restricted, and formal	Better for principles and strategies, access is versatile and innovative
Reasoning computation	Deductive, tedious to program, fast and accurate, poor error correction	Inductive, easier to program, slow, accurate, and good error correction
Sensing	Good at quantitative assessment, poor at pattern recognition	Wide ranges, multifunction, judgment
Perceiving	Copes with variation poorly, susceptible to noise	Copes with variation better, susceptible to noise

In part to help traditional engineers and computer scientists understand the nuances of how humans could interact with a complex system in a decision-making capacity, Levels of Automation (LOAs) were proposed. LOAs generally refer to the role allocation between automation and the human, particularly in the analysis and decision phases of a simplified information processing model of acquisition, analysis, decision, and action phases.[3,4] Such LOAs can range from a fully manual system with no computer intervention to a fully automated system where the human is kept completely out of the loop, and this framework was later expanded to include 10 LOAs (see Table 2).

For LOA scales like that exemplified in Table 2, at the lower levels the human is typically actively involved in the decision-making process. As the levels increase, the automation plays a more active role in decisions, increasingly removing the human from the decision-making loop. This scale addresses authority allocation—for example, who has the authority to make the final decision, and to a much smaller degree, it addresses types of collaborative interaction between the human and computer. Raja Parasuraman and his colleagues later clarified that the LOAs could be applied across the primary information processing functions perception, cognition, and action, and not strictly to the act of deciding but again using the same 10 levels.[4]

Other taxonomies have proposed alternate heuristic-based LOAs, attempting to highlight less rigid and more dynamic allocation structures,[5] as well as address the ability to humans and computers to coach and guide one another. For example, Mica Endsley[6] incorporated artificial intelligence into a five-point LOA scale.

Table 2. Levels of automation.[4]

Automation level	Automation description
1	The computer offers no assistance: human must take all decision and actions.
2	The computer offers a complete set of decision/action alternatives, or
3	Narrows the selection down to a few, or
4	Suggests one alternative, and
5	Executes that suggestion if the human approves, or
6	Allows the human a restricted time to veto before automatic execution, or
7	Executes automatically, then necessarily informs humans, and
8	Informs the human only if asked, or
9	Informs the human only if it, the computer, decides to.
10	The computer decides everything and acts autonomously, ignoring the human.

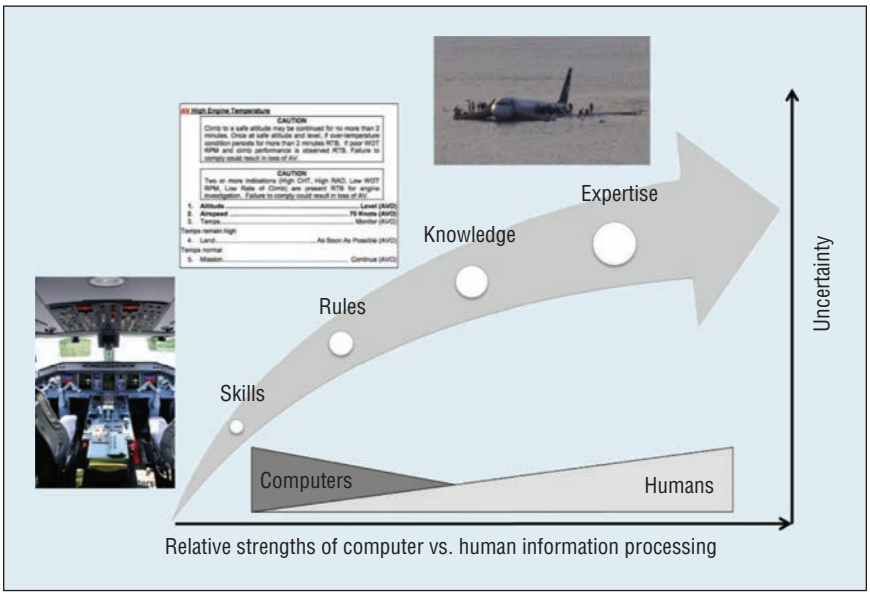


Figure 2. Role allocation for information processing behaviors (skill, rule, knowledge, and expertise) and the relationship to uncertainty.

Such LOA scales have been criticized for their primary focus on an exclusive role and function allocation between humans and computers, and less on the collaborative possibilities between the two.[7] However, as noted previously, engineers and designers of such systems desire some way to determine just when and how to design either exclusive or shared functions between humans and computers, and while imperfect and never intended to be rigid design criteria,[8] the notion of LOAs helped such professionals conceptualize a design space, as well as give them a language to discuss competing design philosophies.

A New Look at an Old Problem

After more than a decade of attempting to train traditional engineers and computer scientists to consider the human early in the design process, in addition to exposing the students to the previously discussed lists and debates surrounding them, the most useful representation I've found that elicits the "aha" moment most educators are looking for is depicted in Figure 2.

First, a map is needed that links information processing behaviors and cognition to increasingly complex tasks, which is best exemplified through Jens Rasmussen's taxonomy of skills, rules, and knowledge-based behaviors[9] (SRK; see Figure 2). One addition to the SRK taxonomy is my representation of uncertainty via the *y* axis. Uncertainty occurs when a situation can't precisely be determined, often due to a lack of or degraded information with potentially many unknown variables. Both external (environmental) and internal (operator performance variability and the use of stochastic algorithms) sources of uncertainty can drive system uncertainty higher.

For Rasumussen,[9] skill-based behaviors are sensory-motor actions that are highly automatic, typically acquired after some period of training. Indeed, he says, "motor output is a response to the observation of an error signal representing the difference between the actual state and the intended state in a time-space environment" (p. 259). This is exactly what controls engineers are taught in basic control theory.

In Figure 2, an example of skill-based control for humans is the act of flying an aircraft. Student pilots spend the bulk of training learning to scan instruments so they can instantly recognize the state of an aircraft and adjust if the intended state isn't the same as the actual state (which is the error signal controls engineers are attempting to minimize.) Once this set of skills is acquired, pilots can then turn their attention (which is a scarce resource,

particularly under high workload), to higher cognitive tasks.

Up the cognitive continuum in Figure 2 are rule-based behaviors, which are effectively those human actions guided by subroutines, stored rules, or procedures. Rasmussen likens rule-based behavior to following a cookbook recipe (p. 261).[9] Difficulties for humans in rule-based environments often come from not recognizing the correct goal in order to select the correct procedure or set of rules.

In Figure 2, in the aviation example, pilots spend significant amounts of time learning to follow procedures. For example, when an engine light illuminates, pilots recognize that they should consult a manual to determine the correct procedure (since there are far too many procedures to be committed to memory), and then follow the steps to completion. Some interpretation is required, particularly for multiple system problems, which is common during a catastrophic failure such as the loss of thrust in one engine. Recognizing which procedure to follow isn't always obvious, particularly in warning systems where one aural alert can indicate different failure modes.

For Rasmussen, the highest level of cognitive control is that of knowledge-based behaviors, where mental models built over time aid in the formulation and selection of plans for an explicit goal.[9] The landing of USAIR 1549 in 2009 in the Hudson River, as Figure 2 shows, is an example of a knowledge-based behavior in that the captain had to decide whether to ditch the aircraft or attempt to land it at a nearby airport. Given his mental model, the environment, and the state of the aircraft, his quick mental simulation made him choose the ditching option.

However, this same accident highlights the importance of the need for a collaborative approach between the human and the machine in that when a complete engine failure occurs in the Airbus 320, the fly-by-wire system automatically trims the plane, computes the ideal glide speed, and readjusts pitch position for landing, which is difficult for pilots to maintain. A single press of the DITCHING button seals the aircraft for water entry. This mutually supportive flight control environment was critical to the successful outcome of this potentially catastrophic event.

I added a fourth behavior to the SRK taxonomy, that of expertise, to demonstrate that knowledge-based behaviors are a prerequisite for gaining expertise in a particular field, and this can't be achieved without significant experience in the presence of uncertainty. So while people can be knowledgeable about a task through repetition, they become experts when they must exercise their knowledge under vastly different conditions. For example, one pilot who has flown thousands of hours with no system failures isn't as much of an expert as one who has had to respond to many system failures over the same time period. Moreover, judgment and intuition, concepts that often make traditional engineers uncomfortable since they lack a mathematical formal representation, are the key behaviors that allow experts to quickly assess a situation in a fast and frugal method,[10] without necessarily and laboriously comparing all possible plan outcomes.

Figure 2 depicts role and function allocation between computers/automation/machines. Such assignments aren't just a function of the type of behavior, but also the degree of uncertainty in the system. It should be noted that these behaviors don't occur in discrete stages with clear thresholds, but rather are on a continuum.

For complex systems with embedded automation, uncertainty can arise from exogenous sources such as the environment—for example, birds in the general vicinity of an airport that *might*, on rare occasion, be ingested in an engine. However, uncertainty can also be introduced from endogenous sources, either from human behaviors or computer/automation behaviors. As evidenced by the Air France 447 crash in 2009 where the pitot-static system gave erroneous information to the pilots due to icing, sensors can degrade or outright fail, introducing possibly unknown uncertainty into a situation. In this case, where the plane crashed because of pilot error, the pilots couldn't cope with the uncertainty since they hadn't gained the appropriate knowledge or expertise.

Skill-Based Tasks

When considering role allocation between humans and computers, it's useful to consider who or what can perform the skill, rule, knowledge, and expertise-based behaviors required for a given objective and associated set of tasks. For many skill-based tasks, like flying an aircraft, automation in general outperforms humans easily. By flying, I mean the act of keeping the aircraft on heading, altitude, and airspeed—that is, keeping the plane in balanced flight on a stable trajectory.

Ever since the introduction of autopilots and more recently, digital fly-by-wire control, computers are far more capable of keeping planes in stable flight for much longer periods of times than if flown manually by humans. Vigilance research is quite clear in this regard, in that it's very difficult for humans to sustain focused attention for more than 20–30 minutes, and sustained attention is precisely what's needed for

flying, particularly for long-duration flights.

There are other domains where the superiority of automation skill-based control is evident, such as autonomous trucks in mining industries. These trucks are designed to shuttle between pickup and drop off points and can operate 24/7 in all weather conditions, since they aren't hampered by reduced vision at night and in bad weather. These trucks are so predictable in their operations that some uncertainty must be programmed into them, or else they repeatedly drive over the same tracks, creating ruts in the road that make it difficult for manned vehicles to negotiate.

For many domains and tasks, automation is superior in skill-based tasks because, given Rasmussen's earlier definition, such tasks are reduced to motor memory with a clear feedback loop to correct errors between a desired outcome and the observed state of the world. In flying and driving, the bulk of the work is a set of motor responses that become routine and nearly effortless with practice. The automaticity that humans can achieve in such tasks can, and arguably should, be replaced with automation, especially given human limitations such as vigilance, fatigue, and the ~ 0.5 second neuromuscular lag present in every human.

The possibility of automating skill-based behaviors (and as we will later see, all behaviors) depends on the ability of the automation to sense the environment, which for a human happens typically through sight, hearing, and touch. This isn't trivial for computers, but for aircraft, through the use of accelerometers and gyroscopes, inertial and satellite navigation systems, and engine sensors, the computer can use its sensors to determine with far greater precision and reliability whether the plane is in stable flight and how to correct in microseconds if there's an anomaly.

This ability is why military and commercial planes have been landing themselves for years far more precisely and smoothly than humans. The act of landing requires the precise control of many dynamic variables, which the computer can do repeatedly without any influence from a lack of sleep or reduced visibility. The same is true for cars that can parallel park by themselves.

However, as previously mentioned, the ability to automate a skill-based task is highly dependent on the ability of the sensors to sense the environment and make adjustments accordingly, correcting for error as it arises. For many skill-based tasks, like driving, vision (both foveal and peripheral) is critical for correct environment assessment. Unfortunately, computer vision still lags far behind human capabilities in many respects, although there's significant research underway in this area. Ultimately, this means that for a skill-based task to be a good candidate for automation, uncertainty should be low and sensor reliability high, which is difficult for many computer vision applications in dynamic environments.

This is why even the most advanced forms of robotic surgery are still just teleoperation, where the doctor is remotely guiding instruments, but still in direct control. Currently robotic surgical tools don't have mature sensors that allow for the closure of the control feedback loop with a high degree of reliability, like those of autopilots. And while some tasks in the driving domain can be automated because of their skill-based nature (like parallel parking), seemingly simple tasks like following the gestures of a traffic cop for a driverless car are extremely difficult due to immature computer vision systems, which don't cope well with uncertainty.

Rule-Based Tasks

As depicted in Figure 2, skill-based behaviors and tasks are the easiest to automate, since by definition they're highly rehearsed and automatic behaviors with inherent feedback loops. Rule-based behaviors for humans, however, require higher levels of cognition since interpretation must occur to determine that, given some stimulus, which set of rules or procedures must be applied to attain the desired goal state.

By the very nature of their if-then-else structures, rule-based behaviors are also potentially good candidates for automation—but again, uncertainty management is key. Significant aspects of process control plants, including nuclear reactors, are highly automated because the rules for making changes are well-established and based on first principles, with highly reliable sensors that accurately represent the physical plant's state.

Path planning is also very rule-based in that given rules about traffic flow (either in the air or on the road), the most efficient path can be constructed. However, uncertainty in such domains makes path planning a less ideal candidate for complete automation. When an automated path planner is given a start and end goal, for the most part the route generated is the best path in terms of the least time (if that is the operator's goal). However, many possibilities exist that automation may not have information about that cause such a path to be either suboptimal or even infeasible, such as in the case of accidents or bad weather.

It is at this rule-based level where there's significant opportunity for humans to collaborate with automation to achieve a better solution than

either could alone. While fast and able to handle complex computation far better than humans, computer optimization algorithms, which work primarily at the rule-based level, are notoriously brittle in that they can only take into account those quantifiable variables identified in the design stages that were deemed to be critical. In complex systems with inherent uncertainties (such as weather impacts or enemy movement), it isn't possible to include a priori every single variable that could impact the final solution.

Moreover, it's not clear exactly what characterizes an optimal solution in such uncertain scenarios. Often, in these domains, the need to generate an optimal solution should be weighed against a satisficing[11] solution. Because constraints and variables are often dynamic in complex environments, the definition of optimal is also a constantly changing concept. In those cases of time pressure, having a solution that's good enough, robust, and quickly reached is often preferable to one that requires complex computation and extended periods of times, which might not be accurate due to incorrect assumptions.

Another problem for automation of rule-based behaviors is similar to one for human selection of the right rule or procedure for a given set of stimuli. Automation will reliably execute a procedure more consistently than any human, but the assumption is that the computer selects the correct procedure, which is highly dependent on the sensing aspect. This is where obstacle detection and avoidance, particularly for driverless cars, is critical. If the automated sensors detect an obstacle, then procedures will be executed for avoidance or braking or both. Indeed, it has been shown that cars equipped with radar can automatically brake much more effectively than a human can.[12] However, the sensing aspect is a significant problem for this futuristic technology, which isn't as reliable in bad weather with precipitation and standing water on roadways.

Knowledge-Based Tasks and Expertise

The most advanced form of cognitive reasoning occurs in domains where knowledge-based behaviors and expertise are required. Coincidentally, these settings are also typically where uncertainty is highest, as Figure 2 shows. While rules may assist decision makers (whether human or computer) in aspects of knowledge-based decisions, such situations are by definition vague and ambiguous and mathematically optimal solutions are unavailable.

It's precisely in these situations where the human power of induction is critical. Judgment and intuition are critical in these situations, as these are the weapons needed to combat uncertainty. Because of the aforementioned brittleness problems in the programming of computer algorithms and the inability to replicate the intangible concept of intuition, knowledge-based reasoning, and especially true expertise, for now, are outside the realm of computers. However, there's currently significant research underway to change this, particularly in the machine learning community—but progress is slow.

IBM's Watson, 90 servers each with a 3.5-gigahertz core processor, is often touted as a computer with knowledge-based reasoning, but people confuse the ability of a computer to search vast databases to generate formulaic responses with knowledge. For Watson, which leverages natural language processing and pattern matching through machine learning, uncertainty is low. Indeed, because Watson leverages statistical reasoning, it can bound answers with confidence intervals.

A more near-term example of human-computer collaboration for knowledge-based medical decision making is the Athena Decision Support System that implements guidelines for hypertension and opioid therapies.[13] This system harnesses the power of computer search and filtering but also allows doctors the ability to guide the computer based on their own experiences.

A limitation of pattern-matching approaches is the overreliance on supervised learning, in that labels must be assigned (typically by humans) for a computer to recognize a pattern. Not only is it possible for humans to introduce error in this process, it raises the question of whether a computer can detect a pattern or event it has never seen before, or that's slightly different than a pattern it has seen before.

There has been increasing interest in using semisupervised and unsupervised machine learning algorithms that don't use labels, and thus generate groups of patterns in absence of such bias. However, with regard to replicating human learning in terms of object recognition, unsupervised machine learning for computers is still quite immature. In a recent major "breakthrough," an unsupervised algorithm was able to cluster and successfully recognize cats in unlabeled images with only 15.8 percent accuracy, which was reported to be an improvement of 70 percent over the current state of the art.[14] For computer vision applications, robust, fast, and efficient perception will be needed before computers can reliably be trusted in perception-based tasks.

With such brittleness, it will be some time before computers can truly begin to approach the expertise of humans, especially in situations

Table 3. Degree of automation as a function of a desired behavior.

Cognitive behavior/task	Degree of automation
Skill-based	Best candidate for automation, assuming reliable sensors for state and error feedback
Rule-based	Possible candidate for automation, if rule set is well-established and tested
Knowledge-based	Some automation can be used to help organize, filter, and synthesize data
Expertise	Human reasoning is superior, but can be aided by automation as a teammate

of high uncertainty. But this isn't to say there's no role for computers in knowledge-based reasoning. Again, this area is ripe for more development in human-computer collaboration. IBM's first commercial application of Watson will be *aiding* nurses and doctors in diagnoses, which falls squarely in the domain of expert decision makers.

While Paul Fitts and his colleagues were perhaps overly focused on mutually exclusive assignment of human and machine roles, their basic premise more than 60 years ago should be interpreted through the lens of collaborative systems and the behaviors that need to be supported. The modified SRK taxonomy presented here isn't meant to be a replacement for earlier role and function efforts, but rather a different lens through which to think about system design. The intent is to provide engineers and computer scientists with a principled framework by which to formulate critical questions, such as the following:

- Can my sensors provide all the data I need at a high enough reliability to approximate trained human skill sets?
- Is there a high degree of uncertainty in either my environment or my sensors, which would necessitate human supervision?
- Can humans augment and improve either sensor or reasoning deficiencies, and how would this occur without overloading the human?
- Can automation reasoning be improved through human guidance and coaching?
- Can automation be leveraged to help the human reduce uncertainty, particularly when knowledge and expertise is needed? The reverse should also be explored in that the human may be able to reduce uncertainty for the automation.

As Table 3 shows, skill-based behaviors are the best candidates for automation, assuming significant sensor performance assumptions can be met, but rule- and knowledge-based reasoning are better suited for human-computer collaboration. Systems should be designed so that humans harness the raw computational and search power of computers for state-space reduction, but also allow them the latitude to apply inductive reasoning for potentially creative, out-of-the-box thinking. As a team, the human and computer are far more powerful than either alone, especially under uncertainty.

In a 2005 competition against the Hydra chess computer, two novices with three computers beat the computer and other grandmasters aided by single computers. Arguably chess is an environment of low uncertainty (particularly for computers that can search a large but finite set of possible outcomes). However, in a real-world and highly uncertain command-and-control environment of one operator controlling multiple robots in a search-and-find task, it has been shown that allowing the human to coach a highly automated system produces results up to 50 percent better than if the automation were left to its own devices.[15] Collaboration between humans and computers, particularly in knowledge-based domains where complementary strengths can be leveraged, hold much future potential.

Last, role and function allocation is as much art as science. The complexity of systems with embedded autonomy supporting dynamic human goals suffers from the "curse of dimensionality."[8] As a result, these systems will never have closed-form solutions and will be intractable from a mathematical perspective. But because of the necessary mix of art and science in designing such systems, both industry and academia should recognize the need for a new breed of engineer/computer scientist. Such a person should have an appreciation for human psychology and performance characteristics, but at the same time understand control theory, Bayesian reasoning, and stochastic processes. ■

References

1. P.M. Fitts, "Human Engineering for an Effective Air Navigation and Traffic Control System," tech. report, *Nat'l Research Council*, 1951.
2. S.W.A. Dekker and D.D. Woods, "MABA-MABA or Abracadabra? Progress on Human-Automation Coordination," *J. Cognition, Technology and Work*, vol. 4, 2002, pp. 240–244.
3. R. Parasuraman, "Designing Automation for Human Use: Empirical Studies and Quantitative Models," *Ergonomics*, vol. 43, 2000, pp. 931–951.
4. R. Parasuraman, T.B. Sheridan, and C.D. Wickens, "A Model for Types and Levels of Human Interaction with Automation," *IEEE Trans. Systems, Man, and Cybernetics—Part A: Systems and Humans*, vol. 30, 2000, pp. 286–297.

5. D.B. Kaber et al., "Adaptive Automation of Human-Machine System Information-Processing Functions," *Human Factors*, vol. 47, Winter 2005, pp. 730–741.
6. M. Endsley, "The Application of Human Factors to the Development of Expert Systems for Advanced Cockpits," *Proc. Human Factors and Ergonomic Soc. 31st Ann. Meeting*, 1987, pp. 1388–1392.
7. Defense Science Board, *The Role of Autonomy in DoD Systems*, US Dept. of Defense, 2012.
8. T.B. Sheridan, "Function Allocation: Algorithm, Alchemy, or Apostasy?" *Intl J. Human-Computer Studies*, vol. 52, 2000, pp. 203–216.
9. J. Rasmussen, "Skills, Rules, and Knowledge: Signals, Signs, and Symbols, and Other Distinctions in Human Performance Models," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 13, no. 3, 1983, pp. 257–266.
10. G. Gigerenzer, P.M. Todd, and T.A.R. Group, *Simple Heuristics That Make Us Smart*, Oxford Univ. Press, 1999.
11. H.A. Simon et al., *Decision Making and Problem Solving*, Nat'l Academy Press, 1986.
12. Insurance Institute for Highway Safety, "More Good News about Crash Avoidance: Volvo City Safety Reduces Crashes," *Status Report*, vol. 48, no. 3, 2013; www.iihs.org/iihs/news/desktopnews/more-good-news-about-crash-avoidance-volvo-city-safety-reduces-crashes.
13. A. Advani et al., "Developing Quality Indicators and Auditing Protocols from Formal Guideline Models: Knowledge Representation and Transformations," *Proc. AMIA Ann. Symp.*, 2003, pp. 11–15.
14. Q.V. Le et al., "Building High-level Features Using Large Scale Unsupervised Learning," *Proc. 29th Int'l Conf. Machine Learning*, 2012; http://icml.cc/2012/papers/73.pdf.
15. M.L. Cummings et al., "The Impact of Human-Automation Collaboration in Decentralized Multiple Unmanned Vehicle Control," *Proc. IEEE*, vol. 100, 2012, pp. 660–671.

Mary (Missy) Cummings is the director of the Humans and Autonomy Laboratory at Duke University. She is also a principle investigator in the MIT Computer Science and Artificial Intelligence Laboratory. Her research interests include human-autonomous system collaboration, human-systems engineering, and the ethical and social impact of technology. Cummings has a PhD in systems engineering from the University of Virginia. She's an IEEE Senior Member. Contact her at m.cummings@duke.edu.

This article originally appeared in IEEE Intelligent Systems, *vol. 29, no. 5. 2014.*

Editor: Carole Goble • carole.goble@manchester.ac.uk

The Semantic Web and End Users: What's Wrong and How to Fix It

David R. Karger • *Massachusetts Institute of Technology*

The Semantic Web's potential to deliver tools that help end users capture, communicate, and manage information has yet to be fulfilled, and far too little research is going into doing so. The author examines the poor state of current tools, argues that the Semantic Web offers a key part of the answer to building better ones, and discusses what needs to change in Semantic Web research to attain that goal.

The Semantic Web has the potential to revolutionize the way that end users capture, communicate, and manage information. But this potential has yet to be fulfilled, and far too little research is going into doing so. Here, I examine the current poor state of user tools, argue that the Semantic Web promises better ones, and discuss how Semantic Web research needs to change if we are to create them.

Amy Voida, Ellie Harmon, and Ban Al-Ani paint a grim picture of the current state of user tools in their work on what they call "homebrew databases."[1] Choosing a specific domain, the authors studied volunteer-driven nonprofit organizations of varying sizes. They interviewed the coordinators responsible for managing information about volunteers, skills, needs, and tasks to learn how they did their jobs. The results were painful. Lacking an application designed specifically to manage their information, these coordinators were forced into a baroque assemblage of Excel spreadsheets, Outlook lists, paper, index cards, and binders. With this mix of tools, they had terrible versioning problems, wasted inordinate amounts of time on data entry and transfer, and struggled to organize, query, and visualize their information.

The coordinators' tasks weren't complicated – they weren't doing big data analytics. Rather, they were trying to answer elementary questions such as "Which volunteers are available for the following activity?" or "What's a summary of all the work this volunteer has done?" Such questions would be trivial for a trained database administrator with a well-maintained SQL database. Unfortunately, few users fit this profile. Although Voida and colleagues studied a specific domain, I've heard similar stories from users in many domains, and believe these problems to be pervasive.

For all of us in databases (and the Semantic Web), this current state of the art is a major embarrassment. We're caught up in grand challenges while everyday users are being defeated by "easy" problems. For those who argue that these users should know better and learn the right database tools to manage their data, famed designer and researcher Don Norman responds:

> When you have trouble with things – whether it's figuring out whether to push or pull a door or the arbitrary vagaries of the modern computer and electronics industries – it's not your fault. Don't blame yourself: blame the designer.[2]

The designer, in this case, is us.

What's the Problem?

The problem we're facing isn't one of information retrieval. Rather, our computers are actively obstructing people from recording and organizing their information. If people can't record it, they certainly can't retrieve it.

 Published by the IEEE Computer Society

In particular, traditional applications are generally developed with a fixed schema in mind. This schema determines both the information that can be stored and how it's presented and manipulated. Users whose information is in a different schema are out of luck — they can't record it properly. (My physical therapist recently described how frustrated she was struggling to enter data about her patients into a new electronic medical record system — until she discovered she could put it all in the comment fields, destroying much of the value of the EMR structure.) Thus, users with nonstandard schemas must use the small set of tools that can handle arbitrary schemas — most frequently, spreadsheets. The homebrew database work highlights how severe the consequences are and even observes that schemas must frequently change on the fly as underlying information needs change.[1]

Fixed-schema applications also pose a severe barrier for users who want to connect information from multiple applications — for example, linking a person in your address book to the music in your media player that person composed. Because these applications are unaware of each other's schemas, they can't do anything with (or even refer to) each other's data. I discuss this issue further elsewhere.[3]

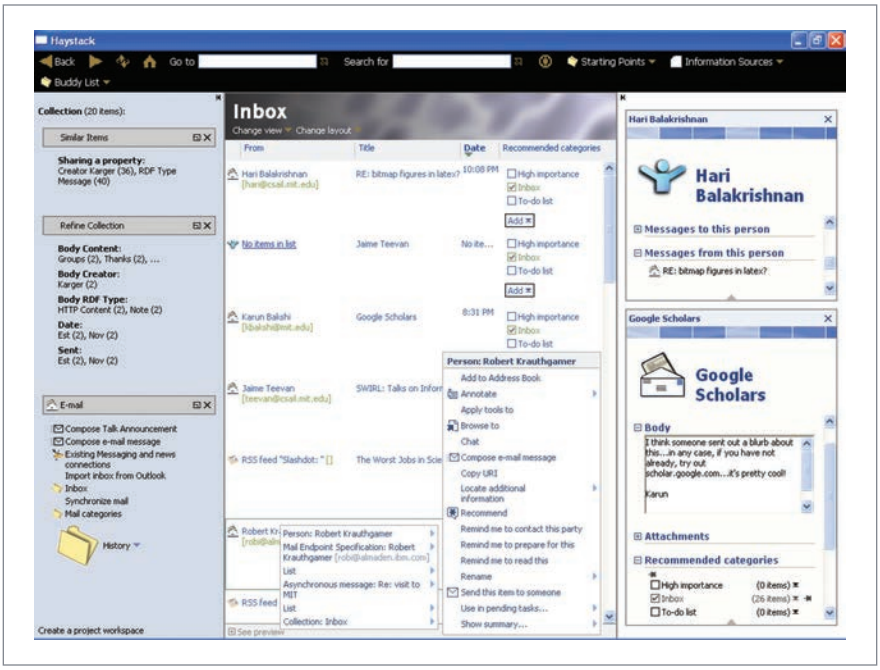


Figure 1. Haystack. The client manages a schema-mixed collection of email messages, news items, and people displayed in a single list, but with each type shown using its own "lens." Facets can filter the collection, and context menus expose operations suited to each item type.

How the Semantic Web Can Help Users

The Semantic Web promises the kind of schema-flexible applications people need. To illustrate, I'll sketch several example tools developed in my group at MIT.

Haystack

Haystack[4] was our first attempt to address the schema-diversity problem (Figure 1). Haystack could store arbitrary, user-defined entities with arbitrary properties and relations to other entities. It also let users customize visualizations of entities. You could create something that looked like a traditional application over whatever schema you decided was useful.

We created the first version of Haystack[5] before the Semantic Web was described.[6] However, it was obvious after the fact that Haystack was a Semantic Web application (more specifically, a semantic desktop; see http://en.wikipedia.org/wiki/Semantic_desktop). When RDF emerged as a Semantic Web-standard data model, we adopted it as the native model for later Haystack versions.[7,8]

Haystack reflects the Semantic Web community's key perspective — the idea of a web of data reflecting vast numbers of distinct schemas. Although the database community has devoted significant effort to data integration, the canonical example has been, for instance, the combination of a few large corporate databases when two companies merge. That community hasn't really addressed the far more anarchic situation of a different schema on each website.

Instead of traditional applications with hard-coded schemas and interfaces, we need applications such as Haystack, whose storage and user interface can effectively present and manipulate information in any schema that the user encounters or creates. This is a challenging task because we tend to rely on knowing the schema to create good user interfaces; however, it's a challenge we can and must meet.

Related Worksheets

Another tool addressing this challenge is Related Worksheets (Figure 2).[9] Eirik Bakke and colleagues decided that because users had voted en masse for spreadsheets, he would find ways to make them better for schema-flexible data management without changing their fundamental nature. His approach improves spreadsheets to better present and navigate the entities and relationships they represent (see http://csail.mit.edu/~ebakke/relspread).

A typical spreadsheet might have one table consisting of university courses (one row per course) referring to another table consisting of course readings (one row per reading) and yet

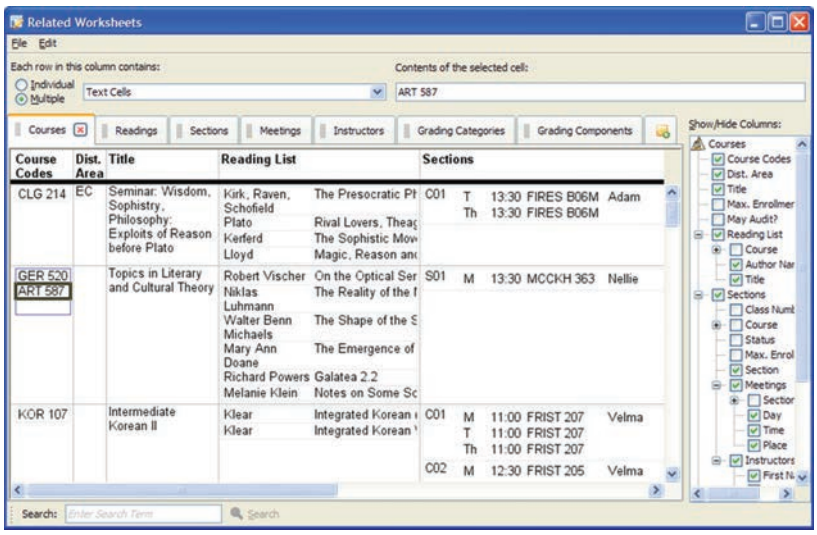


Figure 2. The Related Worksheets system. Here, the system displays information about courses, each with reading and sections. Presentation is nested inside the relevant cells of the course's table. You can see more about Related Worksheets at http://csail.mit.edu/~ebakke/relspread.

Figure 3. The Exhibit framework. Here, the framework presents information about US presidents plotted on a map and a timeline. Facets on the left offer filtering by religion and party, along with text search. You can explore this example at http://simile-widgets.org/exhibit/examples/presidents/presidents.html.

another table of course instructors. In a traditional spreadsheet, this "reference" is just a textual correspondence — there's a cell in the course table naming the title of a reading that's in the readings table.

But if you recognize that the reading is actually an entity, you can do better. First, you can present information about each reading *nested* inside the cell in the course listing table, so you can immediately see more information without having to find it in the readings table. Second, you can "teleport" from a reading shown in the course table to the corresponding row in the readings table, where you can see or modify more data (and, for example, teleport onward to the reading's author). A user study showed that these features can significantly improve users' speed at extracting information from a worksheet. Essentially, these interactions provide the "join" operation that's so powerful in databases and missing in spreadsheets.

Exhibit

Whereas Related Worksheets leverages the spreadsheet metaphor, Exhibit (Figure 3)[10] revisits Haystack's construction of custom GUIs for arbitrary schemas in the context of interactive websites (see http://simile-widgets.org/exhibit/).

Although professional organizations can create fancy data-interactive websites using templates, sorting, faceted browsing, and rich visualizations, most users don't have the programming and database administration skills necessary to do so, and thus tend to publish only text and static images. David Huynh and colleagues recognized that many professional sites fit a common pattern, and that he could define a small extension to the standard HTML vocabulary that sufficiently described these sites.[10] The vocabulary describes common elements such as views (lists, tables, maps, and timelines), facets (for filtering the data shown in the views), and lenses (HTML templates for individual items). Any user can drop these extended HTML elements into a webpage, point them at a data file (CSV, Google spreadsheet, or JSON), and instantly publish an interactive data visualization. To make it even easier, Ted Benson and Adam Marcus created Datapress by integrating Exhibit into Wordpress,[11] so you can "blog your data" using Wordpress's built-in WYSIWYG editor (see http://projects.csail.mit.edu/datapress/).

More than 1,800 exhibits are now online, covering a broad spectrum of

datasets,[12] including ocarinas and failing bridges, European Court for Human Rights cases, pollution measurements in Spain, map stores, classical music composers, sports schedules, prescription drugs, mining information, teacher bonuses in Florida, and an Urdu-English dictionary. Our evaluation and interviews show that Exhibit is meeting its users' need to present interactive data visualizations without complex infrastructure.[12]

To try Exhibit for yourself, you can copy one from the Web and start playing with it. For a more careful introduction, you can view a tutorial at http://people.csail.mit.edu/karger/Exhibit/CAR/HandsOn/.

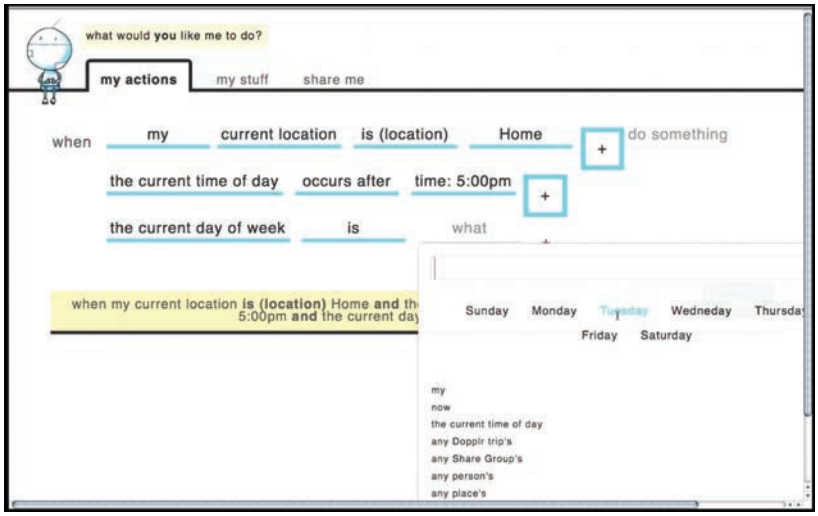


Figure 4. Atomate. Here, the system generates the rule "remind me to take out the trash when I get home on Tuesday evening." The user adds one word at a time; each step uses a dropdown/autocomplete to ensure that only valid words are added. You can view a video of Automate in action via http://doi.ieeecomputersociety.org/10.1109/MIC.2014.109.

Atomate

Beyond visualization, users also need tools to compute over data. Max van Kleek and colleagues built Atomate (Figure 4) to demonstrate how users could author automation rules to reduce the effort needed to handle incoming information streams such as social media.[13] The Semantic Web perspective suggests that these streams could all feed into a common database holding a user's personal and social state. Standing queries could then be used to automatically process this information, reducing manual labor. For example, a user might want to be notified when his calendar shows that a good band is in town and his social media stream reports that a friend is in town, so they can attend the performance together. Or, a user might want to automatically email her secretary when location-tracking data indicates she'll be arriving late.

A key challenge is developing a query language simple enough for everyday users. We designed a *controlled natural language* — a query language that looks like English but is actually unambiguous and easy to parse — as a way to specify *triggers* and *actions* over the properties and values in the user's structured data collection. Triggers are standing queries that are activated by specified configurations of data in the user's database; actions can include messages to the user or other targets, or internal updates to the data. Dropdown menus and autocomplete ensure that the user can create only meaningful queries.

A study of Atomate revealed that users could create meaningful queries when given a specific task, that they recognized the system's general utility, and that they could envision specific ways (particular rules they could write) to use it for their own benefit.[13]

A related tool has since become quite popular. If This Then That (IFTTT; https://ifttt.com) is a website with "channels" that can access various data and social media sites. Users can author recipes that combine triggers over the channels with actions on other channels. IFTTT uses triggers and actions similar to Atomate's but is much more polished and powerful.

IFTTT highlights the benefit of a Semantic Web. At present, the site's creators provide each data source — they must write specialized code to access the specialized API for each Web service they want to turn into a channel. The Semantic Web's vision of a common API for accessing data on websites would obviate the need for specialized development to incorporate each new channel, letting users write triggers and actions over any site they consider useful.

Fixing Semantic Web Research

The examples I've presented provide evidence that Semantic Web approaches can create powerful and usable schema-flexible tools. Given the potential to revolutionize user information management, I'm disappointed with the level of effort the Semantic Web community is investing in it. Research on user applications is almost completely absent from Semantic Web conferences.

Preparing for my keynote at the 2013 European Semantic Web Conference (ESWC; see http://videolectures.net/eswc2013_karger_semantic/), I looked over the conference program. Of the 36 main-track papers, most

were devoted to the Semantic Web's generic underlying technologies — ontology matching (9), entity linking (6), information extraction (6), RDF querying (6), inference (3), and specific RDF ontologies (3). Amidst all this infrastructure, I identified only one paper, by Kaarel Kaljurand and Tobias Kuhn, that described a user application.[14] Indeed, only five of the 36 main-track papers had any screenshot of a working system at all. The 2013 International Semantic Web Conference (ISWC) showed a similar lack of user applications.

ESWC also offers an "in-use track," and it's particularly odd that only two of the six papers in this track came with a screenshot; the other four were narratives about experts putting their data on the Semantic Web — useful, but certainly not user applications.

I now discuss several possible explanations (and treatments) for this lack.

A common risk in academic research is getting too caught up in our hammers (powerful solution techniques) and losing track of the nails (the problems that need solving).

Less Semantic, More Web

The Semantic Web is a new research community, and we might imagine that it was established to tackle a new research direction. But for many ESWC and ISWC papers, this isn't clear. The conference mainstays — knowledge representation, inference, and ontologies — have seen decades of study within the artificial intelligence community. This is because these topics are all essential for the long-term objective of modeling cognition and developing true AI.

This is obviously an important goal, but I question its domination of Semantic Web research. Isn't such work still ongoing in the AI community? Given these problems' fundamental nature, does the fact that we're doing our inferences over Web data rather than, say, an expert system knowledge base change the problem at all? If nothing about this work is specific to the Semantic Web, what's the value of partitioning it from the AI community?

The Semantic Web's major novelty arises from its innovation over the Web, rather than its innovation over semantics. The Web's revolution was in making it easy for everyone to author, manage, and share information. It wasn't really about novel systems — all the pieces already existed. Instead, it was about a novel arrangement of those pieces that empowered users. Structured data can drive this revolution forward, but only if we continue to emphasize the user.

Hammers and Nails

A common risk in academic research is getting too caught up in our hammers (powerful solution techniques) and losing track of the nails (the problems that need solving). I fear this has happened in our community. Back in the early days, we convinced ourselves that a web of structured data would be useful. Now, we're devoting all our energy to inventing an infrastructure for that web. But we never fully worked out just how it would be used. Sure, if we solve AI, then we can send our autonomous agents to do all our work for us on the Semantic Web[6] — but if we solve AI, those agents will be able to understand text and won't need the Semantic Web. Regardless, this solution seems a long way off.

We must do a better job of demonstrating, to ourselves and to others, the Semantic Web's more immediate benefits. We can do this only by showing how it can solve problems that users have right now. Not the generic utopian vision,[6] but nails. We must describe specific user problems and demonstrate specific Semantic Web applications that will solve them now. If we fail to do this — if we create hammers without nails — I doubt we'll ever build the right hammers. Someone else will solve these problems (less well) without using Semantic Web tools, and the Semantic Web will be left behind.

More of our research should start by identifying a current user problem, because then we'll know that there's a real reason to create the solution. Each example application I described previously did this: personal information management and integration under your own schema, making spreadsheets work better, publishing useful interactive data visualizations without programming, and automatically coping with incoming information streams. I'll specifically highlight Atomate[13] because it's actually quite close in spirit to the agents proposed in the Semantic Web visioning paper,[6] but with an important difference. We don't yet know how to build autonomous Semantic Web agents. But we can lower our sights, give people a simple controlled natural language for query specification, and get something that we can build and use right now. Then, we can point to it and say, "Look, if the Semantic Web were widespread, everyone could immediately benefit from something like this!"

Supporting my warning that the Semantic Web might be left behind as irrelevant, consider IFTTT. This tool is tackling the same problem as Atomate,

but solving it without using Semantic Web technology. This makes it inferior in certain ways — it's impossible to integrate new data suppliers and consumers unless the company chooses to do so. However, it's clearly superior in the most important way: it's actually out there, and people are using it to solve their current problems. If we wait too long to offer something better based on the Semantic Web, people will get comfortable with the lesser solution. An O'Reilly post from 2011 speaks approvingly of "API-centric architectures" (such as IFTTT) and disparagingly of the Semantic Web.[15] If we fail to demonstrate our approach's superiority, then theirs will win by default.

As an aside, I'll propose a specific remediation here: next year's Semantic Web Challenge should be to create a Semantic Web version of IFTTT. We could compare different instantiations of the conveniently named SWIFTTT on usability, power, and utility. IFTTT shows that if we actually created a tool that worked, there would be clear demand for it.

Demos

I've been focusing on the papers, but I also want to talk about two pervasive issues I found at the ESWC demo track. The track had plenty of applications, but it suffered from the same problem I wrote about in a blog post two years ago,[16] which initiated the thinking that led to this article. Far too many of these demos called themselves Semantic Web applications because they stored their data in RDF. This makes no sense as the definition of a Semantic Web application. From a user perspective, the particular storage model ought to be invisible. Any system that stores its data in RDF could store it in a SQL database instead, with its user none the wiser (indeed, the user is unlikely to know what RDF and SQL are). Instead, I'll state once more what I consider the key characteristic of Semantic Web applications:

> A Semantic Web application is a Web application whose schema is expected to change.

Few of the applications demoed at ESWC met this description. Rather, the majority seemed to have a fixed schema for their data that was hard-coded into the user interface. The fact that the underlying data was RDF might make it easier to change the schema, but there's no hint of how that change might easily propagate to the user interface.

Evaluation

Another serious problem with the demos was that they lacked evaluations. At first glance, this is understandable: evaluation is the difference between a demo and a research paper. But consider this question: where were all the evaluations of 2012's demos, which should have shown up as research papers at the 2013 conference? There weren't any. This suggests that our builders are stopping at demos and never actually evaluating what they built.

If true, this is a fundamental flaw in our community. As a scientific discipline, we must have evaluation; without it, we can make no progress. Can anyone other than the creator actually use the system? Does anyone see any benefit to doing so? Without positive answers to these questions, our demos tell us nothing. As academics, we aren't building tools for use; we're building tools as experiments that yield data for us to analyze. If we don't gather data, we're wasting our time.

This problem extended to the in-use track as well. I attended the track and, after several of the talks, asked some variant of "How did making this a Semantic Web application make it better than a traditional application?" Some speakers didn't have an answer; others did but hadn't put it in their papers. When prompted, one presenter went on at length about how much easier it had been to build their system[17] using Semantic Web technologies instead of traditional databases. This was great to hear, but this argument was barely made in the paper itself, and no (evaluation-based) proof existed — only opinion. Where was the comparable system that tried to do things the old-fashioned way and was harder? How do we know that Semantic Web technologies were actually better here, as opposed to just being what the developers found most familiar?

As a scientific discipline, we must have evaluation; without it, we can make no progress.

Three of my examples included some form of evaluation. With Related Worksheets, we concentrated on usability: Could users answer questions faster with these than with traditional spreadsheets? Similarly, with Atomate, we asked whether users could actually create rules using our controlled natural language, as well as whether they perceived value in having such a tool. With Exhibit, we studied what kind of data and visualizations users want (and are able) to publish on the Web.[12] We did *lab studies* for Related Worksheets and Atomate, bringing users into our lab to carry out specific tasks we defined for them. These are great for understanding usability, pretty easy to set up, and not too time consuming. Simple metrics such as task correctness and time to completion let us compare different tools. Exhibit warranted more of a *field*

study; we put the tool out in the world and got some adoption, and followed up by asking what users are doing with it. These studies tend to be messier, and they require more time and effort to deploy the tool and analyze the data. They're also more realistic.

I hope I've convinced you that the Semantic Web offers some key insights on schema diversity that are critical to improving peoples' ability to manage information. But we aren't doing the work to turn these insights into useful innovations. We're devoting far too much energy to studying knowledge representation, reasoning, and information extraction that have traditionally appeared in AI conferences, and perhaps should continue to do so. We build applications, but we call them demos and don't evaluate them. Many of them aren't really Semantic Web applications; they're just traditional applications that happen to be storing their data in RDF. We're letting a great opportunity pass us by; let's think about ways to seize it!

Acknowledgments

This article is based on a set of three blog posts that summarize a keynote presentation on "The Semantic Web for End Users" that I gave at the 2013 European Semantic Web Conference. The posts are available at http://haystack.csail.mit.edu/blog/2013/06/05/keynote-at-the-european-semantic-web-conference-part-1-the-state-of-end-user-information-management/. Both the slides and a video of the presentation are available online at http://doi.ieeecomputersociety.org/10.1109/MIC.2014.109 and http://videolectures.net/eswc2013_karger_semantic/, respectively.

References

1. A. Voida, E. Harmon, and B. Al-Ani, "Homebrew Databases: Complexities of Everyday Information Management in Nonprofit Organizations," *Proc. 2011 Ann. Conf. Human Factors in Computing Systems*, 2011, pp. 915–924.
2. D.A. Norman, *The Design of Everyday Things*, Basic Books, 2002.
3. D.R. Karger, "It's All the Same to Me: Data Unification in Personal Information Management," *Personal Information Management*, W. Jones and J. Teevan, eds., Univ. of Washington Press, 2007, pp. 127–152; http://people.csail.mit.edu/karger/Papers/pimchapter.pdf.
4. D.R. Karger, "Haystack: Per-User Information Environments," *Beyond the Desktop Metaphor: Designing Integrated Digital Work Environments*, V. Kaptelinin and M. Czerwinski, eds., MIT Press, 2007, pp. 49–100; http://people.csail.mit.edu/karger/Papers/desktopchapter.pdf.
5. E. Adar, D.R. Karger, and L.A. Stein, "Haystack: Per-User Information Environments," *Proc. 8th Int'l Conf. Information and Knowledge Management*, 1999, pp. 413–422; http://people.csail.mit.edu/karger/Papers/cikm99.pdf.
6. T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," *Scientific Am.*, vol. 284, no. 5, 2001, p. 35; www.scientificamerican.com/article/the-semantic-web/.
7. K. Bakshi and D. Karger, "End-User Application Development for the Semantic Web," *Proc. Semantic Desktop 2005 Workshop, Int'l Semantic Web Conf.*, LNCS 2870, 2005; http://people.csail.mit.edu/kbakshi/SemDesk%20-%20Final.pdf.
8. D.R. Karger et al., "Haystack: A General Purpose Information Management Tool for End Users of Semistructured Data," *Proc. Conf. Innovative Database Research*, 2005, pp. 13–26; http://www-db.cs.wisc.edu/cidr/cidr2005/papers/P02.pdf.
9. E. Bakke, D. Karger, and R. Miller, "A Spreadsheet-Based User Interface for Managing Plural Relationships in Structured Data," *Proc. 2011 Ann. Conf. Human Factors in Computing Systems*, 2011, pp. 2541–2550; www.mit.edu/~ebakke/research/related_worksheets_chi2011.pdf.
10. D. Huynh, R. Miller, and D.R. Karger, "Exhibit: Lightweight Structured Data Publishing," *Proc. 16th Int'l Conf. World Wide Web*, 2007, pp. 737–746.
11. E. Benson et al., "Talking About Data: Sharing Richly Structured Information through Blogs and Wikis," *The Semantic Web: ISWC 2010*, LNCS 6496, 2010, pp. 48–63.
12. E. Benson and D.R. Karger, "End-Users Publishing Structured Information on the Web: An Observational Study of What, Why, and How," *Proc. 32nd Ann. ACM Conf. Human Factors in Computing Systems*, 2014, pp. 1265–1274.
13. M. Van Kleek et al., "Atomate It! End-User Context-Sensitive Automation using Heterogeneous Information Sources on the Web," *Proc. 19th Int'l Conf. World Wide Web*, 2010, pp. 951–960; http://people.csail.mit.edu/emax/papers/atomate-www2010-camera.pdf.
14. K. Kaljurand and T. Kuhn, "A Multilingual Semantic Wiki Based on Attempto Controlled English and Grammatical Framework," *The Semantic Web: Semantics and Big Data*, LNCS 7882, pp. 427–441; http://eswc-conferences.org/sites/default/files/papers2013/kaljurand.pdf.
15. A. Jhingran, "Six API Predictions for 2012," O'Reilly Radar blog, 19 Dec. 2011; http://radar.oreilly.com/2011/12/api-predictions-2012.html.
16. D. Karger, "Two Funny Things at the 2012 International Semantic Web Conference," Haystack blog, 26 Nov. 2012; http://haystack.csail.mit.edu/blog/2012/11/26/two-funny-things-at-the-2012-international-semantic-web-conference/.
17. L.M. Garshol and A. Borge, "Hafslund Sesam — An Archive on Semantics," *The Semantic Web: Semantics and Big Data*, LNCS 7882, 2013, pp. 578–592; http://eswc-conferences.org/sites/default/files/papers2013/garshol.pdf.

David R. Karger is a professor of electrical engineering and computer science in the Computer Science and Artificial Intelligence Laboratory at the Massachusetts Institute of Technology. His research interests include algorithms, machine learning, information retrieval, personal information management, networking, peer-to-peer systems, coding theory, the Semantic Web, and human-computer interaction. Karger received a PhD in computer science from Stanford University. He's an ACM Fellow. Contact him at karger@mit.edu.

This article originally appeared in IEEE Internet Computing, *vol. 18, no. 6. 2014.*

Knocking Down Strawmen

Vinton G. Cerf • *Google*

Much has been made of the terms "net neutrality" and "openness" in connection with the Internet. Many who argue against the importance and preservation of these properties interpret these terms in ways that enable their arguments. For the most part, I consider many of these interpretations to be deliberately fabricated so as to make them easy to knock down and thus belittle proponents of neutrality and openness. The Internet has always been characterized as a best-efforts system that doesn't make guarantees as to delivery, latency, or even sequence of arrival. The protocols that are used to implement it are intended to be resilient in the face of a variety of failures.

Among the misinterpretations of neutrality, we find "every packet must be treated identically." In fact, the intent is more along the lines of "equal potential for impairment." The Internet's philosophy is that it will do the best it can to service any arriving packet, but it's free to drop the packet or shape the traffic to protect network resources, deal with congestion, and provide fair access to its facilities. Naysayers will argue that the network's implementation isn't neutral unless every packet receives identical treatment. This is a strawman argument. The network should be essentially indiscriminate with regard to origin or traffic destination, and should supply its best efforts to deliver packets while considering limits to capacity. User choice should be protected so that network access providers don't accidentally or intentionally create limits on users' choices of application or destination within some framework of fairness.

Any particular Internet network does have finite capacity and will be subject to congestion. Moreover, network operators must respond to congestion to limit the load to available capacity. At the Internet's edges, one limiting factor is the bandwidth of the access network. It isn't a violation of neutrality to limit users at the Internet's edge to a maximum bandwidth for which they might have paid. Neutrality doesn't mean that providers can't charge for more usage. During periods of congestion, we might expect that users paying for more capacity would be permitted to consume a larger portion of what's available than those paying for less peak capacity.

Some Internet access providers limit the total number of bytes users can send or receive in the course of a billing period (for example, a month). Such practices are a weak proxy for shaping traffic to fit within the immediately available bandwidth at the network edge. The network is somewhat insensitive to sending traffic at rates that are low relative to the available bandwidth. The limiting factor is the rate of transmission more than the total number of bytes sent or received. A terabyte in a month puts far less pressure on available network resources than a terabyte in 10 seconds.

Recall that packet switching takes advantage of statistical multiplexing. Each user has momentary access to the maximum bandwidth of his or her access channel, but all that capacity is available to others. The system assumes that statistical sharing produces a better average experience for all users until the system approaches congestion, at which point providers must understandably impose traffic shaping and fair allocation of capacity.

Many of the arguments against neutral treatment are based on the finite nature of the capacity of the Internet or its access networks. In fact, over time, the Internet has grown dramatically in capacity, not only in scale but also in the maximum data rates at the edges and in the core. The Arpanet had only 50 Kbytes/s to offer in the core, but today's optical fiber networks operate

 Published by the IEEE Computer Society

at 100 Gbytes/s — a factor of two million! The Internet's neutral posture has allowed any new and all existing applications equal opportunity to access this vastly increased capacity. A rising tide raises all boats, and so it is with the Internet.

None of these arguments are intended to defeat reasonable efforts to protect users' interests (for instance, fair access, protection from harm, or network availability), although we might reasonably have different views as to how these protections should be implemented. Some protections might be better implemented elsewhere besides the Internet's basic packet-carrying layers.

Regarding openness, it seems fair to say that the Internet's architecture and philosophy have permitted it to absorb and use just about every communication technology that developed since its conception in 1973. The IP layer is insensitive to the medium by which packets are carried and is also ignorant of the applications that generate and consume those packets. Some have argued that the Internet could do a better job of adapting its resources if it could know in advance what an application's usage pattern might be (low-latency games, high-capacity video streaming, high-rate file transfers, latency-tolerating email, and so on). Although you could make a case for this, the convergence of all types of data transfer within a single application (as with the Web) begs classification questions. The intent of the open Internet mantra is simply that the Internet won't dictate what applications are mounted or used but, rather, will do the best it can to serve them. This posture has led to an unbounded number of new applications, none of which had to get permission from all ISPs in the world to be mounted and offered or used. Of course, some will work poorly if the available capacity is inadequate to the task. Openness also implies freedom to invent new protocols to support new applications. The layering of the protocol stack has enabled horizontal expansion in protocol space at each layer.

Here is perhaps a crux of the debate. For some, the present limits on available capacity should dictate implementation of mechanisms to guarantee that particular applications will work. For others, myself included, the opportunity to gain access to the Internet's increasing capacity without rigid classification is the incentive for inventing new and useful applications, new protocols, and new transmission technology.

We can add another interpretation of the term openness: freedom to speak and to hear. Many of us believe the Internet and the Web have enhanced these freedoms immeasurably. To these freedoms, I think we must also add freedom from harm, which leads to understandable concerns about and desires to protect users from malware, denial-of-service attacks, identify theft, and a host of other ills that are found in the Internet environment. It's fair to say that protecting these freedoms is a shared responsibility among the Internet's users and providers, law enforcement, and the regulatory frameworks used worldwide. The Internet is a global system, and protection of user freedoms will require the cooperation and harmonization of national and international practices, to the extent practicable.

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

This article originally appeared in IEEE Internet Computing, *vol. 18, no. 6, 2014.*

Editors: Lisa Avila and Mike Bailey

Advanced Display Technologies

Recent advances in visual display solutions are enabling users to experience digital content in new ways and in a variety of locations. Whether they serve as the focal point of user interaction or to augment real or virtual environments, the products we review here provide innovative approaches to content consumption and interaction. Application areas for such technologies include gaming, classrooms, museum exhibitions, trade shows, marketing, simulation and training, and more.

Spherical Projection Display

Computer graphics display devices are usually flat, or perhaps curved, but many of the things we want to display, such as planets and moons, are spherical. Pufferfish has announced a series of spherical projection displays, the PufferSpheres, just for applications like these.

PufferSpheres come in a variety of sizes, from 600 mm (2 feet) to 4000 mm (13 feet). Some are hard surfaced and some are inflatable; the inflatable ones are less expensive. The PufferSpheres can function at any orientation—they can sit on a table, hang from a ceiling, or mount on a wall.

Such displays are mesmerizing to watch. Figure 1 shows our planet as it really is, not according to a particular flat-map projection. For interactive applications, some of the PufferSpheres have a touch-sensitive surface. For custom content creation, there is a Unity3D SDK to develop 3D displays within the Unity environment and a plug-in for Adobe AfterEffects. For more information, see http://pufferfishdisplays.co.uk.

Virtual Gaming Environment

The Cyberith Virtualizer is a 3D virtual reality

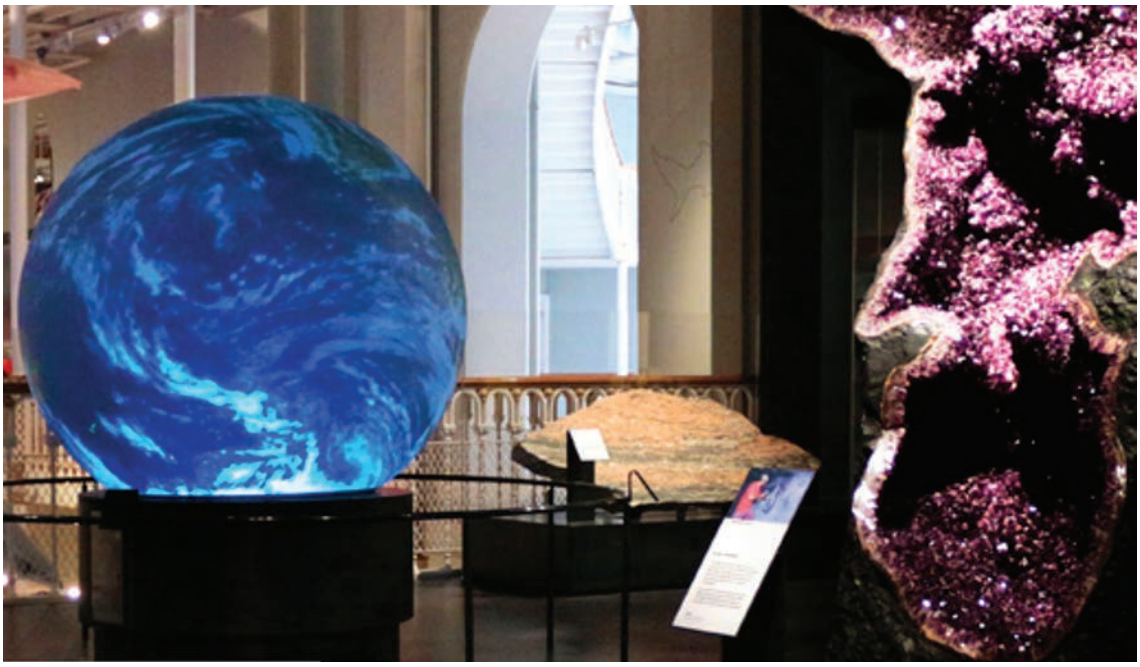


Figure 1. Pufferfish PufferSphere spherical projection display. This PufferSphere XL display at the National Museum of Scotland shows the Earth. (Photo courtesy of Pufferfish.)

Published by the IEEE Computer Society

gaming interface device that allows a user to duck, twist, and run within a virtual environment, without actually going anywhere. The device consists of a 100 cm (39 inch) diameter low-friction base plate and three 100 cm (39 inch) high pillars. The user is suspended from a harness that is attached to the three pillars. The harness can support a weight up to 120 kg (265 pounds). In the harness, users can twist a full 360 degrees, enabling them to "advance" in any horizontal direction (see Figure 2) as well as walk, run, crouch, jump, or sit.

The Virtualizer connects to your computer with a simple USB plug. Through the USB, the Virtualizer draws power and is able to look like a game controller so that no special drivers are required. Thus, for many games, the Virtualizer is plug-and-play. For those who want to do custom programming, a Virtualizer SDK is available.

For more information, see http://cyberith.com.

Figure 2. Cyberith Virtualizer 3D virtual reality gaming interface device. The harness allows the user to twist a full 360 degrees and thus "advance" in a game in any horizontal direction. (Photo courtesy of Cyberith.)

Augmented Reality Glasses

The Epson Moverio BT-200 augmented reality glasses are yet another step in the advancement of wearable computing. These lightweight, relatively inexpensive glasses project binocular images transparently so that the viewer can see both the image and the world beyond. The images are projected using micro-projection and see-through imaging technology such that the viewer perceives them as a 200 cm (80 inch) screen.

Head-tracking is accomplished with a built-in 3D tracker. Communication takes place over either Wi-Fi or Bluetooth. The battery is rechargeable and is rated for six hours of average use. The Epson Moverio BT-200 comes with a head set, controller, 4 Gbyte microSDHC card, detachable earphones, AC adapter, and carrying case.

It has long been recognized that augmented reality is a valuable technology. Many of us have been anxious to live in a world that is capable of annotating itself, that is, one in which computer-generated imagery can enhance our understanding of what is happening around us. The growth of augmented reality has long suffered from glasses that are too expensive to buy and too cumbersome to use. The Epson Moverio BT-200 looks like a good step in the direction of increased use of augmented reality (see Figure 3).

The Moverio SDK, developer support information, and the Moverio App are available in the Moverio Apps Market (https://moverio.epson.biz/). For more information, see www.epson.com/cgi-bin/Store/jsp/Landing/moverio-bt-200-smart-glasses.do.

Figure 3. Epson Moverio BT-200 augmented reality glasses. The lightweight smart glasses project transparent images using micro-projection and see-through imaging technology. (Photo courtesy of Mike Bailey.)

Contact department editors Lisa Avila at lisa.avila@kitware.com and Mike Bailey at mjb@cs.oregonstate.edu.

This article originally appeared in
IEEE Computer Graphics & Applications, *vol. 35, no. 1, 2015.*

Strong Hiring Outlook for 2015

Tech professionals aiming to find or change jobs in 2015 can take heart. Record-breaking hiring is expected for the first half of the year.

In its semiannual survey of nearly 800 tech-focused recruiters, career website Dice found that 75 percent anticipate hiring more tech professionals in the first half of 2015 than during the previous six months—an all-time high. Nearly three quarters of these companies plan to expand hiring by more than 10 percent during the period.

According to Dice president Shravan Goli, "The year ahead looks bright for tech professionals. In high demand and with hiring managers looking to hire a substantial number of new hires, tech professionals really have strong negotiating power,"

As employees increasingly leave companies for better jobs elsewhere, hiring managers see candidates asking for more money, and counteroffers are becoming the norm. Robert Half Technology expects substantial average starting salary increases for six jobs in particular this year.

- *Mobile applications developer.* Professionals skilled in developing apps for tablets and smartphones can expect a 10 percent hike in starting pay, with an average salary between $107,500 and $161,500.
- *Big data engineer.* Organizations launching big data initiatives will need experienced engineers able to translate business objectives into data processing workflows. Big data engineers can anticipate a 9.3 percent boost in starting pay in 2015, with average salaries ranging from $119,250 to $168,250.
- *Wireless network engineer.* Professionals skilled in wireless network design, implementation, and optimization will be in demand as more companies launch internal infrastructure projects to support rising mobile device and wireless technology use. Wireless network engineers can expect a 9.1 percent bump in base compensation, with average starting salaries between $99,000 and $137,500.
- *User experience (UX) director.* Organizations need creative leadership to ensure consistent user experience across web and mobile properties that aligns with business strategy and brand identity. Experienced UX directors can anticipate average starting salaries between

Cont. on p. 70

 Published by the IEEE Computer Society

SENIOR TERADATA WAREHOUSE CONSULTANT. Analyze and develop logical database designs, data models, and relational database definitions across multiple computing environments utilizing Teradata and SAP platforms and tools. Identify and implement new uses of information technologies to meet strategic objectives. Prepare business requirements reports, recommendations, feasibility studies and cost justification statements. Analyze and design the Teradata Warehouse Architectures to support the insurance business including the extraction, transformation and loading of data from Mainframe policy administration systems. Apply to: M. Gagne, MIP F110, Massachusetts Mutual Life Insurance Company, 1295 State St, Springfield, MA 01111.

SAP SECURITY ANALYST. Provide SAP security support to managers, perform security reviews and testing of the SAP systems, participate in audit and compliance activities. Perform security upgrades, manage transports, and troubleshoot access issues. Build and maintain SAP user roles across the SAP landscape including ECC, BI, PI, SM and Enterprise Portal. Apply to: M. Gagne, MIP F110, Massachusetts Mutual Life Insurance Company, 1295 State Street, Springfield, MA 01111.

ENGINEERING

QUANTUMSCAPE CORPORATION is accepting resumes for the position of Sr. Member of Technical Staff in San Jose, CA. Responsible for maintaining an atomic layer deposition (ALD) fabrication tool, running it to produce thin films of specified characteristics and properties, and measuring film properties. Mail resume to QuantumScape Corporation, Staffing Department, 1730 Technology Drive, San Jose, CA 95110. Must reference Ref. Code SMTS-CC.

SIEMENS PLM SOFTWARE INC. has the following openings: Software Engineer Adv/UGS159 in Milford, OH to design, develop, modify, & implement software programming for products. Requires up to 5% domestic travel to client sites. Software Engineer Adv/UGS165 in Troy, MI to develop data mgmt. solutions for Mechatronics in Teamcenter. Email resumes to PLMCareers@ugs.com & refer to Job title/code of interest. EOE.

SOFTWARE ENGINEER, San Mateo, CA sought by MDOTM, F/T, Dsgn & implmt user profiling, web systms for advertising platform in PHP/Java, Participate in reqmts review, functional spec, Technical Dsgn/Algorithms; participate in brainstorming sessions & contribute ideas to our tech, algorithms & products; help improve systm qlty through writing unit tests, automation & performing code reviews. Masters in Comp Sci or Info Systms & 2 yrs S/ware Engg exp. Knowl of: 1. Object Oriented Dsgn & coding skills in PHP/Java; 2. Web dvlpmt in Java/PHP; 3. Web svcs dsgn & dvlpmt; 4. SQL dbases. Resume to careers@mdotm.com.

NUVIEW TECHNOLOGIES INC. has the following opening in Orlando, FL:Programmer Analyst:- Analyze user requirements, procedures & problems to improve existing systems using Java based technologies. Test, maintain, monitor computer programs & systems. Knowledge of Databases & SQL. Req. 2 yrs exp. Send resume to Jobs@Nuviewtech.biz. This position will involve working in unanticipated locations.

ADVERTISER INFORMATION • FEBRUARY 2015

Advertising Personnel
Debbie Sims
Advertising Coordinator
Email: dsims@computer.org
Phone: +1 714 816 2138
Fax: +1 714 821 4010

Chris Ruoff
Sales Manager
Email: cruoff@computer.org
Phone: +1 714 816 2168
Fax: +1 714 821 4010

Advertising Sales Representatives (display)
Central, Northwest, Far East:
Eric Kincaid
Email: e.kincaid@computer.org
Phone: +1 214 673 3742
Fax: +1 888 886 8599

Northeast, Midwest, Europe, Middle East:
Ann & David Schissler
Email: a.schissler@computer.org, d.schissler@computer.org
Phone: +1 508 394 4026
Fax: +1 508 394 1707

Southwest, California:
Mike Hughes
Email: mikehughes@computer.org
Phone: +1 805 529 6790

Southeast:
Heather Buonadies
Email: h.buonadies@computer.org
Phone: +1 201 887 1703

Advertising Sales Representative (Classified Line & Jobs Board)
Heather Buonadies
Email: h.buonadies@computer.org
Phone: +1 201 887 1703

BMC SOFTWARE INC.. has an opening for Sr. Technical Support Analyst in Lexington, MA to install, configure, test & troubleshoot business software application releases in a customer support environ. Email resumes to Olivia_Delgado@bmc.com refer to Req# 15000025.

ENGINEERING. TRANSCRIPTIC, INC. is accepting resumes for the position of Member of Technical Staff in Menlo Park, CA. Utilize mechanical engineering knowledge to gather technical requirements, design, prototype and fabricate lab automation robotics for biological research. Mail resume to Transcriptic, Inc., Staffing Department, 3565 Haven Street, Menlo Park, Suite 3, CA 94025. Must reference Ref. MTS-JS.

COMPUTER SYSTEM ANALYST for a Medical and Surgical Clinic located in Anahuac, TX. Applicant must possess a Bachelor's Degree in Computer Information System. Job duties are to design and develop computer systems by configuring hardware & software & devise ways to apply existing systems to tasks. Must have knowledge of VB6.0, MySQL, Windows Server2000/ 2003, Revenue Mgt & Practice Mgt System & EMR system. Compensation based on experience. Respond by resume only to: Dr. Leonidas Andres, Job Code CSA001, Andres Medical Clinic, PO Box 1470, Anahuac, TX 77514.

.NET APPLICATION DEVELOPER (Chicago, IL) Write, review or re-write Microsoft Excel add-in prgm to support daily financial analyst & processing; Enhance, upgrade, existing financial reports publishing web application using Classic ASP, ASP.NET, JavaScript/JQuery, AJAX, XML and SQL. Reqs: MS in comp sci., s/ware tech. or closely rel., 24 mths exp. as comp prgmr/analyst. Resumes to Daniel Smereczynski, VP, First Analysis Securities Corp, One South Wacker Dr., Ste 3900, Chicago, IL 60606.

PURDUE UNIVERSITY. Tenure-Track/Tenured Faculty Positions: The Department of Computer Science at Purdue University is entering a phase of significant growth, as part of a university-wide Purdue Moves initiative. Applications are being solicited for 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 early in October 2014, and will continue until the positions are filled. The Department of Computer Science offers a stimulating and nurturing academic environment with active 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 related discipline, be committed to excellence in teaching, and have demonstrated excellence in research. 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

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

Facebook, Inc. currently has the following openings in **Menlo Park, CA** (various levels/types):

Product Designer (3214J)

Design, prototype, and build new features for Facebook's website or mobile applications.

Localization Project Manager (2686J)

Deliver all projects on time across all supported locales to align with product releases - create and execute on the localization schedule.

Facebook, Inc. currently has the following openings in **Seattle, WA** (various levels/types):

Industrial Designer (IDJ)

Contribute to all aspects of product development process from product research and early concept development through engineering and transfer to manufacturing.

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

background check will be required for employment. Purdue University is an EEO/AA employer fully committed to achieving a diverse workforce. All individuals, including minorities, women, individuals with disabilities, and protected veterans are encouraged to apply.

SPRUCE TECHNOLOGY, INC., Clifton, NJ based IT firm, is seeking multiple candidates for following positions: **Sr. Programmer Analyst:** Develop & write computer programs to store, locate, & retrieve specific docs, data, & info; Analyze user needs & software req's to determine feasibility of design w/in time & cost constraints; Design, develop & implement the next gen platforms using tools & software w/back-end databases to provide an integrated management system. Will use a combination of T-SQL, PL/SQL, ProClarity, SSRS/SSAS/SSIS, Microstrategy, Informatica, OLAP, C++, ASP, Java Script, VB.Net, MDX. Masters in Engg (any) CS, Science (any), Comp Application w/ 1 year of related exp is req'd. Will accept Bachelor's degree w/5 yrs of related exp as equal to Master's degree. **Sr. Business Systems Analyst:** Design, test & conduct tech writing of software apps; Analyze, Plan & Develop Business Programs; Manage resources in accordance w/project schedule; Gather & synthesize business req's & translate the business req document; Design methodology & programs to ensure that the project deliverables meet industry best practices & standards; Review & Modify Software programs to fulfill desirable accuracy & reliability of programs; Coordinates & link computer systems within an organization to increase compatibility so information can be shared. Use Java, J2EE, JSP, Websphere, Weblogic, Oracle, SQL Server, PL/SQL. Bachelor degree in B. Admin (any), Science (any) or Comp. Sci. w/5 yrs of related occupation exp is req'd. Will accept combination of degrees that is equal to Bachelors. **Sr. Software Engineer:** Gather & analyze tech req's; Design & develop software programs; Develop interaction models & user flow diagrams; Resp for software devel life cycle incl analysis, design, devel, implementation, & support; Write database queries & involve in testing the application. Will use a combination of J2EE, JavaScript, Ajax, Servlets, Beans, Hibernate, Springs, ApacheAxis, JBossSeam, JBossCache, JBuilder, Eclipse, Flex Builder. Bachelor'sin Engg (any) CS, Science (any), Computer Application w/5 yrs of exp in the related field is req'd. Any combination of education equal to Bachelors is acceptable. **Database Engineer:** Resp for analysis, modeling, design, devel & implementation of relational database and data warehousing systems; Conceptualize & communicate enterprise data architecture frameworks for global enterprises; Resp for programming & developing database packages, functions, & triggers. Will use a combination of ETL tools DataStage, SSIS, Informatica, AbInition, TeraData, Business Objects, MicroStrategy, SQL*Loader, ODBC, Toad, Visual Basic, Java, Korn Shell and Unix Shell Scripting. Bachelor in Engg (any), CS, Science (any), Computer Application w/2 yrs of exp in the related field is req'd. Any combination of education equal to Bachelors is acceptable. **Sr. SAP Analyst:** Analyze, develop & maintain of software apps by using SAP HCM; Devel functional business process specs, document technical solutions & maintain business process procedures; Setup SAP-Success Factors Compensation activities; Configure SAP R/3 system for complete testing & internal order functionality; Involve in all phases of project life cycle using ASAP methodology; Involve in data migration by using SAP LSMW tools; Work on enhancements/interfaces and custom development with EDI interface. Will use SAP R/3, HPQC, SAP Service Market Place, Success Factors Cloud Support Portal. Bachelor in Science (any), B. Admin (any), CIS with 5 years of related occupation experience is required. Will accept combination of degrees that is equal to Bachelors. Apply with 2 copies of resume to Spruce Technology Inc., 1149 Bloomfield Avenue, Suite G, Clifton, NJ 07012.

UCF CENTER FOR RESEARCH IN COMPUTER VISION. Multiple Assistant Professor Positions. CRCV is looking for multiple tenure-track faculty members in the Computer Vision area. Of particular interest are candidates with a strong track record of publications. CRCV will offer competitive salaries and start-up packages, along with a generous benefits package offered to employees at UCF. Faculty hired at CRCV will be tenured in the Electrical Engineering & Computer Science department and will be required to teach a maximum of two courses per academic year and are expected to bring in substantial external research funding. In addition, Center faculty are expected to have a vigorous program of graduate student mentoring and are encouraged to involve undergraduates in their research. Applicants must have a Ph.D. in an area appropriate to Computer Vision by the start of the appointment and a strong commitment to academic activities, including teaching, scholarly publications and sponsored research. Preferred applicants should have an exceptional record of scholarly research. In addition, successful candidates must be strongly effective teachers. To submit an application, please go to: http://www.jobswithucf.com/postings/34681 Applicants must submit all required documents at the time of application which includes the following: Research Statement; Teaching Statement; Curriculum Vitae; and a list of at least three references with address, phone numbers and >>email address. Applicants for this position will also be considered for position numbers 38406 and 37361. UCF is an Equal Opportunity/Affirmative Action employer. Women and minorities are particularly encouraged to apply.

ENGINEERING. Zscaler, Inc. is accepting resumes for the position of Senior QA Engineer in San Jose, CA. Lead a project team of engineers to design, develop and test company software products, including protocols, such as TCP/IP, HTTP, SSL and Encryption technologies, to provide cloud security. Mail resume to Zscaler, Inc., Staffing Department, 110 Baytech Drive, Suite 100, San Jose, CA 95134. Must reference Ref. SQE-GB.

VirginiaTech
Invent the Future

Computer Science–Dept. Head

The Department of Computer Science at Virginia Tech seeks applications from creative and visionary leaders for the position of Department Head. The Department Head's principal responsibility is to provide leadership and management of the department's programs, faculty, staff, and students. This entails leadership of departmental programs and administrative responsibility for planning, fiscal management, human resources, and communication within the department. The Department Head is expected to advance the research and teaching missions of this prominent department, nurture interdisciplinary collaborations, and work to achieve strategic goals in both the department and university. The successful candidate will be located at the Blacksburg, VA campus and lead a department with faculty there and in the National Capital Region campus (**www.ncr.vt.edu**). Faculty in NCR are located in Falls Church, VA as well as in the Virginia Tech Research Center (**www.ncr.vt.edu/arlington**) in Arlington, VA.

Doctoral degree in Computer Science or a closely related field; demonstrated intellectual leadership and administrative skills in an academic/university environment or equivalent.

For a full description and to apply, please see: **http://jobs.vt.edu** for posting **TR0140155**. Inquiries should be directed to Dr. Dennis Kafura, Search Committee Chair (**kafura@cs.vt.edu**, 540.231.5568).

Virginia Tech is an AA/EEO employer; applications from members of underrepresented groups are especially encouraged.

SUNFIX TECH reqs (1) Data Warehouse BI Specialist to provide data management, data warehousing & business intelligence solutions. ETL/DataWarehousing Exp req. 2) Sr. Programmer Analyst to analyze user req's, design & devel custom software apps. Java/J2EE exp req. Positions require Masters (Engng/Comp. Sci) +1 year exp; BS +5 years exp can be substituted for the MS degree req. Any combination of foreign edu equiv to a US MS or BS will be considered. Position req's extensive travel to client sites. Send res to Sunfix Technologies, Inc. at 15 Corporate PL South, Suite 140, Piscataway, NJ 08854.

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

MEMBER OF TECHNICAL STAFF/ SERVER DEVELOPER. (San Francisco) S/ware co. seeks individual to join our server dvlpmt team & create scalable, high qlty, & secure server s/ware. Duties will incl: analyzing the product reqmts in order to come up w/best possible solution, while ensuring product reliability; bldg infrastructure to manage metadata for millions of user's social contacts & handling them in a way which maintains user's privacy; bldg highly scalable, realtime applics using the appropriate prgmg lang (e.g. Java, Go, Shell script, etc), & dsgn patterns, as reqd by the task; dsgng complex domain models which can be easily persisted to our long term data stores, incl various data store backends (e.g. MySQL, Apache Lucene, or other NoSQL backend); dvlp tools to automate & simplify both dvlpmt & production tasks; dsgn & dvlp end-to-end svcs & frameworks that can be used by our IOS & Android clients, or on the web; write comprehensive test cases & run them on Continuous Integration tools (e.g. Bamboo, Hudson, etc.) to ensure code qlty & reliability. Reqmts: Master's deg in Comp Sci or rltd disciplines, + 2 yrs exp or Bachelor's deg + 6 yrs. Alternate combo of education/exp will be considered. Send resume to Humin Inc., 655 Montgomery St, Ste 1950, San Francisco, CA 94111. Fax 212-619-5210.

EMPLOYMENT OPPORTUNITIES (UNIONDALE, NY): SR. PROGRAMMER ANALYSTS: Develop & write computer programs to store locate & retrieve specific documents, data & info; Implement web applications using ASP.NET w/VB.NET on an oracle back-end & writing PL/SQL. Develop & direct software system testing & validation procedures, programming, & documentation; Compile & write documentation of program development & subsequent revisions, inserting comments in the coded instructions. Will use ASP.NET, SQL Server, VB.NET, Visual Studio. Bachelor Deg in Engineering (any), Comp. Sci., Science (any), MIS w/ 5 yrs of exp in any related field is req'd. **SR. SYSTEMS ADMINISTRATORS:** Review & Test all EDI transaction processing output & other applications systems; Supervise EDI data flow & managing operation; Setup & configure EDI Systems & coordinate w/Utilities for tests & live production systems; Perform data backups & disaster recovery operations; Administer, configure & maintain system applications & network environment; Troubleshoot & resolve hardware, software, or other network & system problems. Master's Deg in Comp. Sci., Engineering (any), Science (any), MIS is req'd. Resumes to EC INFOSYSTEMS, INC. 50 Charles Lindbergh Blvd. Suite 411, Uniondale, NY 11553.

ENGINEERING. PIVOTCLOUD INC. is accepting resumes for the following positions in Sunnyvale, CA: Principal Software Engineer (Ref. PSE-SS): Architect, dsgn & dvlp compliance, content monitoring, supervision, surveillance, legal discovery, & litigation support applications on mult client-side devices, w/ extensive use of C/C++/Objective C/Java/C#, Windows Phone,.Net, SQL Server, SharePoint Server, Exchange Server, & other rltd Microsoft technologies. Software Engineer (Ref. SE-LK): Architect & dsgn high scale distributed cloud services. Dsgn & implem't security S/W for data storage & data mgmt in the cloud. Position based at co. headquarters in Sunnyvale, CA, or can be home based in continental U.S. Mail resume to PivotCloud, Inc., Staffing Dept, 530 Lakeside Dr, Ste 180, Sunnyvale, CA 94085. Must reference Ref. Code.

LEAD ARCHITECT. (Atlanta, GA & various unanticipated locations throughout U.S.): Plan engineering activities of global bus. integration sols. projs. REQ Bach. deg., (or for. deg. equiv.), in Mgt. Info. Syst. or Comp. Sci. & 4 yrs. exp. in prov. intgr., dev. & consulting for B2B sfwr apps. Stated exp. must incl. 2 yrs. exp. in customer-site implem. of sfwr projs. & app. intgr. Ext. dom. / int'l travel is req. (50%). (Job Code "CG".) **PROJECT MGRS.** (mult. openings) (Atlanta, GA): Dir. R&D & plan engineering activities of global bus. integration sols. projs. REQ Master's deg. in Industrial Engin. Aca. crswrk in Probabilistic Models; & Deterministic Optimization is req. Ext. dom. / int'l travel is req. (50%). (Job Code "PM".) Send resume w/Job Code "PM" or "CG" to Dan Marischuk, Seeburger, Inc., 1230 Peachtree St, NE, #1020, Atlanta, GA 30309.

NISUM TECHNOLOGIES. has multiple openings for the following positions at its office in Brea, CA. *Technical Lead: Design, develop and test software systems. *Sr. QA Engineer: plan and conduct analysis, inspection, design, test and/or integration to assure quality of projects. *QA Engineer: Assist in analysis, design, test and/or integration of projects. *Sr. Application Developer: design, develop, maintain & test software applications/systems. Analyze customer requirements & custom design systems as needed. *Database Administrator: Install, maintain, administer & troubleshoot databases. *Programmer Analyst: Analyze, develop & write codes to implement system applications. Job requires min. of M.S./foreign equiv. + exp., M.S./foreign equiv., or B.S./foreign equiv. + exp.. Education/Experience requirements vary depending on position level/type. Travel/relocation required. Send resume and salary history & position applied for to: Nisum Technologies, 500 S. Kraemer Blvd., Brea, CA 92821. Attn: H.R. Manager.

CLOUDERA, INC. is recruiting for our San Francisco, CA office: Software Engineer: design & implement large distributed systems that scale well – to petabytes of data & 10s of 1000s of nodes. Mail resume w/ job code SE-DA to: Attn.: HR, Cloudera, 1001 Page Mill Rd., Bldg. 2, Palo Alto, CA 94304.

CLOUDERA, INC. is recruiting for our Palo Alto, CA office: Software Engineer: maintain a thorough understanding of the Hadoop eco-system that Cloudera ships with CDH. Create test cases for testing CDH components with Hue & run those tests to validate different CDH releases in a timely manner SE-DV to: Attn.: HR, Cloudera, 1001 Page Mill Rd., Bldg. 2, Palo Alto, CA 94304.

RIVERBED TECHNOLOGY, INC. has opening for Network and Application Performance Engineer (Job # 2015-1816) in Bethesda, MD (telecommuting from anywhere in U.S. is acceptable). Duties: Plan, design, deploy, operate and optimize service provider/large enterprise infrastructures. For more info and to apply, go to http://www.riverbed.com/us/careers/ and reference the Job#.

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

Information Systems Manager [REQ#9FRU9X]. Mng team of Architects & DBAs for Apple Information Sys.
Software Engineer Applications [Req#9D32DG]. Research & dev large-scale Cloud-based productivity app suite across mult platforms.
Software Development Engineer [Req#9JPMUV]. Conduct SW testing for all iOS apps.
Systems Design Engineer [Req#9DE444]. Resp for multi-radio co-existence perform eval, data analysis, & design optimization on a wide variety of projects for various wireless techs.
Software Development Engineer [Req#9CYPUL]. Develop & maintain test frameworks for HTTP Live Stream & progress dwload techs for iOS & OSX platforms.
Software Development Engineer [Req#9AQL67]. Design & dev sw & add new customer facing features.
Software Quality Assurance Engineer [Req#9AYSSN]. Develop, run & maintain SW tests for communication systems.
Software Engineer Applications [Req#9H4NPQ]. Design & develop end-to-end advanced analytic solutions on core Apple data sets.
Software Engineer Applications [Req#9H3TFZ]. Design & dev the next generation of Apple's Employee Syst platform & suite of products. Travel req'd: 25%.
Software Quality Assurance Engineer [Req#9K6SJ3]. Create & execute test plans for vid encoder, pre-processor, post-processor & vid algorithm module used in various Apple apps.
Information Systems Engineer [Req#9GJ364]. Design, dev, implement & maintain EAI, internal Cloud, DB & J2EE app syst.
SAP Performance Analyst [Req#9GFVWS]. Perform SAP performance testing, troubleshooting, and tuning.
Systems Design Engineer [Req#9GPR5M]. Perform RF System Design Validation and Debug for wireless telecommunication systems.
Computer Vision Research Engineer [Req#9DCSRS]. Design & dev algorithms & SW for Computer Vision sys.
Software Development Engineer [Req#9GJRL6] Lead team of build engs to sup ongoing SW builds of major rel of iOS & OS X operate sys.
Software Development Engineer [Req#9DNPSZ] Des, dev & imple SW for routing service.
Systems Design Engineer [Req#9DGTRY] Dev, opt & debug calibration & perform verif stations. Travel req. 20%.
Systems Engineer [Req#9E52CY] Build & trouble comp server sys.
Software Development Engineer [Req#9CXSES] Des & dev natural lang process tech for local Apple prod into int'l markets & help scale art intel apps including Siri to new lang.
Software Engineer Applications [Req#9P92BS] Define & evan OS X/1OS diag data analytic sys.
Operations Project Engineer, New Products [Req#98F49S]. Lead OEM oper. team. Plan & execute dev builds & new prod ramps. Travel req'd 30%.
ASIC Design Engineer (IC Packaging Engineer) [9E5VV6]. Perform IC pckg mech simulation & characterization.
Software Development Engineer [9GXRKD]Serve as a member of the core op sys network team. Provide SW dev in comp network tech.
Hardware Development Engineer [Req #9D2VXQ]. Respon for design & develop of new prods. Design display tech including new pixel & circuit. Travel req 20%
Software Development Engineer [Req#9F4SZH] Design complex distribute sys w/ an emphasis on high avail & perform.
Systems Architect [Req #9AYN8F]. Respon for providing tech guide to drive exec of multi projects. Dsgn high perform, scalable sw sols to supp high demand of daily trans.
Software Engineer Applications [Req#9J2TXF]. Perfrm data mining & rsrch of data w/ focus on developing automated processes & tools to surface actionable data.
Software Engineer [Req#9F4SQW]. Dsgn & dvlp app interfaces, database interfaces, & SW layer abstractions. Lead dvlpmnt of server-side SW components, data persistence, & caching components.
ASIC Design Engineer [Req#9CYUG2]. Prfrm regular STA runs & anlyz, triage & deliver timing results. Validate netlist, parasitics, constraints or other input collateral for quality.
Systems Design Engineer [Req#9E8Q4M]. Dsgn & evaluate Radio Frequency System for iPhone and iPad.
Software Development Engineer [Req#9CCN8P]. Design & develop Network Protocol Software for desktop & mobile platforms.
Software Architect [Req#9GWPE4]. Dsgn, dvlp, & deploy SW apps for Data-warehousing / Business Intelligence Projects in the Apple IST Marketing Dept.
Reliability Engineer [Req#9E23BV]. Guide design teams in creating reliable designs for novel HW technologies.
Hardware Development Engineer [Req#9AE2X8]. Dvlp new designs, panel processes & optics. Lead engineering investigation on new concepts. Travel Required 25%.
Hardware Development Engineer [Req#9DF2GM]. Respon for the design & develop of baseband power mgmt. sols for future wireless products.
Software Development Engineer [REQ#99WSUF]. Perform initial design & dev of apps. Share expertise in app & framework devlpmt.
Product Design Engineer [Req# 9EZW4P] Dsgn & dvlp materials & processing for consumer electronics products.
Software Development Engineer [Req#9FSPTK]. Manage end-to-end lifecycle(s) of complex Wireless functionality in Apple products. Write test plans, test cases, develop automation & ad-hoc testing.
Software Engineer Applications [Req #9GASLJ]. Respon for test sw that forms foundation for iCloud products & srvcs.
Software Development Engineer [Req#9K2VU3] Dev, des & implmt, architect for SW components. Write code used in maps search svs.

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

It's work that matters. It's what we do at Symantec. Symantec is the world leader in providing solutions to help individuals and enterprises assure the security, availability, and integrity of their information. In essence, we protect the free flow of information in a connected world. As the fourth largest independent software company in the world, Symantec has operations in more than 40 countries with 475 out of Fortune's global 500 companies using our solutions.

People look to us to safeguard the integrity of their information, ensuring it is secure and available. Achieving this ambitious goal is only possible through the combined efforts of the innovators and visionaries that Symantec continuously attracts. Symantec draws the very best people with a variety of backgrounds, experiences and perspectives and provides them with a work environment where uniqueness is valued and empowered. The creative people we attract help define the spirit of innovation at Symantec. Symantec is proud to be an equal opportunity employer.

We currently have openings for the following positions (multiple positions/various levels/types):

Columbia, Maryland

Software Engineers (SWEMD115) Responsible for analyzing, designing, debugging and/or modifying software; or evaluating, developing, modifying, and coding software programs to support programming needs.

Culver City, California

Computer Systems Analyst (CSACC115) Analyze engineering, business and/or other business intelligence issues for application to Symantec solutions; and/or provide operational support in the development and implementation process of computer software applications, systems or services.

Database Administrator(DBCC115) Support both development and production environments. Responsible for primary application/database and working with DBA team. The applications supported by these databases are from a wide variety of vendor provided to in-house developed apps.

IT Infrastructure Specialist (ITSCC115) Manage large complex infrastructure by designing, planning, and implementing complex infrastructure systems. Establish and recommend policies and standards on system use and services and automate monitoring or periodic preventative maintenance processes; or Analyze user requirements, procedures, and problems to automate or improve existing systems and review computer system capabilities, workflow, and scheduling limitations.

Knowledge Engineer (KECC115) Build, maintain and use knowledge-based systems. Collect and analyze data for projects and departmental needs, create reports, scorecards and dashboards to analyze performance and results of projects and on-going business. Work with and support projects that require the collection and analysis of data.

Security Infrastructure Administrator (SIACC115) Perform system and database administration for the ongoing maintenance of security network architecture and systems. Maintain multiple production, development and QA SQL Server environments.

Software Engineers: (SWECC115) Responsible for analyzing, designing, debugging and/or modifying software; or evaluating, developing, modifying, and coding software programs to support programming needs.

Software QA Engineers: (SQACC115) Responsible for developing, applying and maintaining quality standards for company products. Develop and execute software test plans. Analyze and write test standards and procedures

Herndon, Virginia

Computer Systems Analyst (Mgr, Ops) (1648.1513) Prioritizes & tracks MSS customer analysis issues. Requests and analyzes new functionality.

Engineering Managers (EMVA115) Direct and supervise team of engineers (QA and/or development teams); Develop standards for products and/or oversee development and execution of software and/or analysis of test results.

Houston, Texas

Software QA Engineers (1648.1589) Responsible for developing, applying and maintaining quality standards for company products. Develop and execute software test plans. Analyze and write test standards and procedures.

Lindon, Utah

Senior Technical Education Staff Member (1648.609) Deliver skills/technical training for Symantec products and conduct trainer events. Develop curriculum design for big picture view of learning and development. Must be available to work on projects at various, unanticipated sites throughout the United States. May Telecommute.

Submit resume to JOBADS@symantec.com . Must reference position & code listed above. EOE.
For additional information about Symantec and other positions visit our website at http://www.symantec.com.

Symantec™

It's work that matters. It's what we do at Symantec. Symantec is the world leader in providing solutions to help individuals and enterprises assure the security, availability, and integrity of their information. In essence, we protect the free flow of information in a connected world. As the fourth largest independent software company in the world, Symantec has operations in more than 40 countries with 475 out of Fortune's global 500 companies using our solutions.

People look to us to safeguard the integrity of their information, ensuring it is secure and available. Achieving this ambitious goal is only possible through the combined efforts of the innovators and visionaries that Symantec continuously attracts. Symantec draws the very best people with a variety of backgrounds, experiences and perspectives and provides them with a work environment where uniqueness is valued and empowered. The creative people we attract help define the spirit of innovation at Symantec. Symantec is proud to be an equal opportunity employer.

We currently have openings for the following positions (multiple positions/various levels/types):

Mountain View, California

Agile Product Owner (1648.1995) Work with a cross-functional team of UI developers, architects, designers and QA testers to ensure that the website is intuitive and designed with a long lifespan in mind in a mobile-friendly way.

Computer Systems Analyst (CSAHQ115) Analyze engineering, business and/or other business intelligence issues for application to Symantec solutions; and provide operational support in the development and implementation process of computer software applications, systems or services.

Data Scientists (DSHQ115) Design, develop, and program methods, processes, and systems to consolidate and analyze unstructured, diverse "big data" sources to generate actionable insights and solutions for client services and product enhancement.

Data Science Director(DADHQ115) Develop and code software programs, algorithms, and automated processes to cleanse, integrate, and evaluate large datasets from multiple disparate sources.

Engineering Manager(EMHQ115) Direct and supervise team of engineering (QA and/or development teams). Develop standards for products and/or oversee development and execution of software and/or analysis of test results. Plan, design, develop and implement processes.

Network Systems Engineer (NSEHQ115) Design, architect, and maintain network systems. Perform tasks related to network engineering, planning, and configuration.

Oracle Database Administrator (1648.2419) Responsible for handling Oracle EBS 11i and R12 environments along with related environments which depends on ERP systems. Support both development and production environments.

Product Managers (PDMHQ115) Develop company market requirements for technical products or product lines, including product strategy definition, requirements analysis, and/or pricing.

Product Marketing Manager (PMMHQ115) Develop product marketing strategy to drive product demand. Plan the launch of new products and releases and manage the cross-functional implementation of the plan.

Production Specialist (1648.1283) Involvement in all phases of eCommerce data entry, collection, generation and validation. Assist in data processing and maintenance of the catalog merchandizing & pricing rules for the online stores.

Search Engine Marketing (SEM) and Display Managers (SEMHQ) Responsible for search marketing strategies and plans for a portfolio of regions within the global team. Assist with developing pay-per-click (PPC) account strategies and roadmaps.

Senior Principle Project Manager Specialist (1648.1119) Manage the scrum activities of two eBusiness Teams. Serve and support the Product Owner and Development Team in their quest to do everything possible to delight customers.

Software Engineers (SWEHQ115) Responsible for analyzing, designing, debugging and/or modifying software; or evaluating, developing, modifying, and coding software programs to support programming needs.

Software QA Engineers (SQAHQ115) Responsible for developing, applying and maintaining quality standards for company products. Develop and execute software test plans. Analyze and write test standards and procedures.

Web Developers (WEBHQ115) Design and develop web applications and websites; create and specify architectural and technical parameters. Designing and implementing of the PC Tools website (and associated websites.

San Francisco, California

Program Manager (1648.2444) Work closely with engineering members, managers and leads, product managers, ensure rapid execution and on time, high quality delivery of complex Data Loss Prevention (DLP) projects.

Program Manager (Product Lifecycle Engineer) (1648.303) Participate in all software product development life cycle activities. Move software products through the product development cycle from design and development to implementation and testing.

Software Engineers (SWESF115) Responsible for analyzing, designing, debugging and/or modifying software; or evaluating, developing, modifying, and coding software programs to support programming needs.

Submit resume to JOBADS@symantec.com . Must reference position & code listed above. EOE.
For additional information about Symantec and other positions visit our website at http://www.symantec.com.

Symantec™

Cont. from p. 62

$110,500 and $178,000, up 6.8 percent from 2014.

- *Interactive creative director.* To execute successful interactive marketing and advertising campaigns, companies look for creative leaders adept at coordinating efforts among designers, writers, and art directors. Skilled interactive creative directors can expect average starting salaries to increase 5.7 percent in 2015, ranging from $100,500 to $180,250.
- *Experienced web designer.* Web designers able to ensure that companies' Internet/intranet sites and digital communications reflect business goals, objectives, and identity will also be in demand. Web designers with five or more years' experience can expect to earn between $80,000 and $112,500 on average, a gain of 4.8 percent over last year.

There's a slight downside to all this. Facing a shortage of skilled talent, companies are taking longer to fill positions, according to Dice, so hiring managers are becoming more discerning. Three-quarters are now looking for candidates with at least 6 to 10 years' experience.

Still, mobile, big data, and security will help drive a highly competitive hiring market over the coming months. As Robert Half senior executive director John Reed notes, "Companies recruiting for workers in software development, security, and networking may find it especially challenging to recruit skilled talent." Job seekers with this edge should be sitting pretty.

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

Cont. from p. 72

Choices and Responsibilities

Responsibility for restricting and filtering Mason's access to that data flow will fall to my wife and myself. But how will we be able to censor his information? How can I be sure to choose with certainty when he learns about all that's available there? If you've ever taken a wrong turn on the Internet (on purpose or accidentally), you know that the "other stuff" is often right next to the "good stuff."

We're left with several options—none of which seems like a great solution. We could lock up all the computers in the house (oh, yeah, and the phones and tablets too); we could make sure our devices have proper parental controls in place (see http://support.apple.com/en-us/HT201304); we could try going back to the good old days of VHS and smoke signals.

Or maybe at the interface level we could discover a way to incorporate some sort of intelligent guide—some AI-powered interface assistant monitoring Mason's every Internet connection.

Kids Find Loopholes

Unfortunately, kids are smart. They figure out loopholes. They exploit weaknesses in rules. You'd have to hardwire into the interface mechanism some control point where we could tell the Internet, sure, show our kids everything—*except* that "other stuff" (however we as individual parents might define it). But even if we *could* make the interface this benign AI personal assistant, how long would it be before our kids were conducting social engineering on the hapless Suri/Watson/Cortana/Now descendent?

When Mason's older, is he going to talk about how late night one night, when he was 8, after chatting with his Internet Guide and interacting with Net ('cause he'll be hip enough to call him/her/it "Net"), he found people disproving that Santa exists?

Damn. The Internet is going to ruin Christmas, isn't it?

But that's okay, because the Internet is one of those awesome, scary things that could potentially change our very foundations, our very capacity for knowledge. We can learn about things so terrifyingly quickly that it's hard to think about the not-so-distant past where we had no idea what was going on that day (or month) on the other side of this big ball spinning around a celestial gas explosion at 108,000 km/h, or 67,000 mph. (I cite as my source for that fact, what else? The Internet.)

What's Next?

The things my peers "earned" through learning have all been uploaded, reposted, and documented on all corners of the Web, so learning for my son and his generation could devolve into simple acts of creative searching or conversations with a personal Internet guide.

Our single information source (in all its multivolume hardcover beauty on a living room shelf) is morphing into a collection of digital pages with source verification determined by millions of armchair researchers' and general know-it-alls' cognitive surplus.

And it's scary to think of my kid wandering those unkempt, virtual halls, not to mention the digital trail he'll leave as he peruses page after page filled with facts and information—and the inevitable misinformation as well.

Which is why we've had Mason start what will be an ever evolving conversation with the Internet early. He needs to be able to speak its language—and vice versa.

And just in case this should ever end up on the Internet, I'll ask my own question.

"Hey, Internet. How're you doing?"

Brian Kirk is IEEE Computer Society's associate manager for editorial product development. Contact him at bkirk@computer.org.

"Hey, Internet! How're You Doing?"

Brian Kirk

One evening recently at dinner, baffled by another of our four-year-old's pretty much daily attempts to understand the essential nature of reality, my wife and I turned to a trusted information repository. And since we usually try to back up our claims with a verifiable source, we told him that to answer his question, we'd ask the Internet.

"What's the Internet?"

I explained that the Internet is a magical collection of information, provided by people all over the world, and … other stuff. Then I did the only logical thing I could think of: I asked Mason what he'd ask the Internet himself if he could pose one question. He thought a moment.

"Hey, Internet! How're you doing?"

Talking to the Internet

What's crazy is that, soon, the Internet may well answer that question (at least, once we can combine enough natural language processing and vocal recognition to enable a child to ask the question matter-of-factly and receive a conversational response). What will it say? And how would the Internet come up with her/his/its response? Maybe it'll analyze data congestion before saying, "Constipated." Maybe it'll be a slow day in the content-curation space, so it'll say, "Hungry."

This also made me think about the wonderful, amazing world that my son is growing up in.

We had a *World Book Encyclopedia* set when I was growing up, and I loved that thing. (Of course, not every family could afford the *World Book*. We were lucky.) Encyclopedia publishing schedules meant the info was great—if the fact that it was two or more years old didn't make a difference.

Other than *World Book*, my learning as a kid was limited to what was available at the library, what was showing on television at any given time, and what was reported in the Sunday paper. Bottlenecks to data retrieval like these will be inconceivable to Mason. He'll be able to learn about and understand things at a much earlier age than I did. My son's "encyclopedia" is the sum of all the information in the world, constantly updated, and … other stuff.

Cont. on p. 71

 Published by the IEEE Computer Society