Computer AUGUST 2011 The IBM PC: 30-Year Retrospective//Cloud Computing Volume 44 Number 8

Innovative Technology for Computer Professionals

Computer

AUGUST 2011

http://www.computer.org

THE IBM PC: 30-YEAR RETROSPECTIVE

CS ELECTION CANDIDATES, P. 65

IEEE PRESIDENT-ELECT CANDIDATES, P. 75

IEEE

IEEE computer society

Innovative Technology for Computer Professionals
Computer

Circulation: *Computer* (ISSN 0018-9162) 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 90720-1314; voice +1 714 821 8380; fax +1 714 821 4010; **IEEE Computer Society Headquarters**, 2001 L Street NW, Suite 700, Washington, DC 20036. IEEE Computer Society membership includes $19 for a subscription to *Computer* magazine. Nonmember subscription rate available upon request. Single-copy prices: members $20.00; nonmembers $99.00.

Postmaster: Send undelivered copies and address changes to *Computer*, IEEE Membership Processing Dept., 445 Hoes Lane, Piscataway, NJ 08855. Periodicals Postage Paid 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.

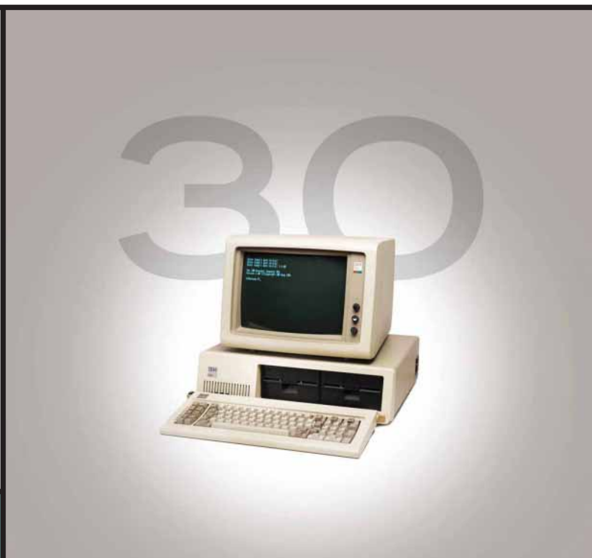
Computer Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page

Innovative Technology for Computer Professionals

Computer

www.computer.org/computer

CONTENTS



ABOUT THIS ISSUE

It's been 30 years since the first IBM PC changed computing history. We in the computer industry have a notoriously short collective memory, but now and again it's wise to take a look back, to honor the pioneers for their contributions, and also to see where decisions made a long time ago still resonate today. In this issue, IBMers Dave Bradley and Mark Dean give the inside story of the creation of the IBM PC, Edward Bride discusses early PC software, and Intel's Gurbir Singh talks about the chips inside.

See www.computer.org/computer-multimedia for multimedia content related to the features in this issue.

COVER FEATURES

COMPUTING PRACTICES

PERSPECTIVES

For more information on computing topics, visit the Computer Society Digital Library at www.computer.org/csdl.

IEEE Computer Society: http://computer.org
Computer: http://computer.org/computer
computer@computer.org
IEEE Computer Society Publications Office: +1 714 821 8380

Flagship Publication of the IEEE Computer Society

August 2011, Volume 44, Number 8

Computer Highlights Society Magazines

The IEEE Computer Society offers a lineup of 13 peer-reviewed technical magazines that cover cutting-edge topics in computing, including scientific applications, design and test, security, Internet computing, machine intelligence, digital graphics, and computer history. Select articles from recent issues of Computer Society magazines are highlighted below.

IEEE Software

Making manifest, syndicating, and then governing a system's architecture facilitates understanding, reasoning about, and transforming that system with intention. This premise holds true for new systems as well as legacy ones, exploratory systems as well as production ones. Watson, IBM's reasoning system, is such a system—it's both new and exploratory. Managing its architecture has considerable payoff. Read "The Soul of a New Watson" by Grady Booch in the July/August issue of *Software*.

IEEE Intelligent Systems

To relieve the continually increasing stresses on drivers and reduce the number of accidents, current intelligent vehicle research is attempting to understand and model driver behaviors. "Toward Cognitive Vehicles," in the May/June issue of *IS*, surveys recent work on cognitive vehicles that model drivers in a stimuli-decision-reaction mode and, on the vehicle system side, improve perception, suggestion, and function delegation of the traffic environment. The authors illustrate the relationships between recent models and methods and list-related research challenges, while introducing applications of the driver-cognition models in intelligent vehicle control systems.

IEEE Computer Graphics and Applications

Beginning to emerge is a new, robust generation of physics-based animation approaches that maintain a standard of visual quality as high as data-driven synthesis. And even with the Holy Grail of control principles that describe human motion still a mystery, the animation research community continues to forge its own path. Researchers have learned that they don't need to solve the problem of biological control, nor do they need to throw out the advantages of animator control and motion capture. Instead, current research aims to find the best of all worlds, judiciously combining physics with human-motion examples, animator input, or both. A July/August special issue of *CG&A* brings together four examples of the innovations in this exploding area.

Computing in Science & Engineering

The National Academy of Sciences recently released a National Research Council study that confirmed what many experts have been observing—computing is undergoing a radical change. The study summarizes why the processing speed of individual computer chips will no longer increase dramatically each year, discusses the implications of this for computing, and outlines what is needed to continue to improve computing performance in the future. Author Douglass Post examines the NRC study in "The Future of Computing Performance" in the July/August issue of *CiSE*.

IEEE Security & Privacy

Building Dependability, Reliability, and Trust

Smartphone manufacturers and mobile-OS vendors are selling record numbers of units, and thousands of developers are forming communities around each of the popular smartphone platforms. In the May/June issue of *S&P*, "Secure Software Installation on Smartphones," an overview of iOS, Android, BlackBerry, and Symbian security frameworks, includes a novel classification of third-party-application installation models. It also describes how controlled app marketplaces fit in the smartphone security ecosystem.

IEEE Pervasive Computing

Mobile and Ubiquitous Computing

For many years, pervasive computing research has explored the potential benefits of creating a connection

Published by the IEEE Computer Society

0018-9162/11/$26.00 © 2011 IEEE

between the Internet's virtual world and the physical world we live in. The Near Field Communication (NFC) standard might, at last, be the technology that makes this vision—sometimes referred to as the Internet of things—a ubiquitous reality. Read "Near Field Communication" by Roy Want in the July-September issue of *PvC*.

IEEE Internet Computing

Internet Computing's July/August issue features a special theme on Web technology and personal health records. Unlike electronic health records that only healthcare providers can manage, the consumer controls PHRs. For example, Google Health and Microsoft HealthVault both let users manage certain aspects of their care. Guest editors Chimezie Ogbuji of Case Western Reserve University, Karthik Gomadam of Accenture Technology Labs, and Charles Petrie of Stanford CS Logic Group describe PHR systems as a potentially disruptive technology and introduce three articles that address some of the challenges and opportunities they present.

IEEE micro

As smart mobile devices become pervasive, vendors are offering rich features supported by cloud-based servers to enhance the user experience. Such servers implement large-scale computing environments, where target data is compared to a massive preloaded database. In "CogniServe: Heterogeneous Server Architecture for Large-Scale Recognition" in the May/June issue of *Micro*, a team of authors from Intel Labs and the Seoul National University of Science and Technology looks at recent advances in mobile/cloud technology. CogniServe is a highly efficient server for large-scale recognition that employs a heterogeneous architecture to provide low-power, high-throughput cores, along with application-specific accelerators.

IEEE MultiMedia

Since Sutherland's SketchPad in 1961 and Xerox's Alto in 1973, computer users have long been acquainted with technologies other than the traditional keyboard for interacting with a system. Recently, with the desire for increased productivity, seamless interaction, immersion, and e-inclusion of people with disabilities, along with progress in fields such as multimedia, multimodal signal analysis, and HCI, multimodal interaction has emerged as an active field of research. Read "Using Modality Replacement to Facilitate Communication between Visually and Hearing-Impaired People" in the April-June issue of *MultiMedia*.

IT Professional

TECHNOLOGY SOLUTIONS FOR THE ENTERPRISE

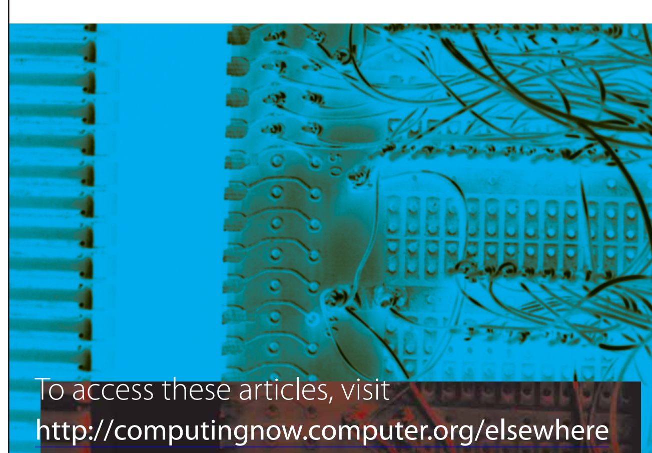
Healthcare organizations must test their network infrastructures for disaster recovery and emergency mode operations, yet most can't afford to operate the complicated protocols needed for safe testing. The Rapid Adjustable Network architecture offers a solution. Author James Teeter of Indiana State University looks at the new architecture in "Flexible Medical-Grade Networks" in the May/June issue of *IT Pro*.

IEEE Design&Test of Computers

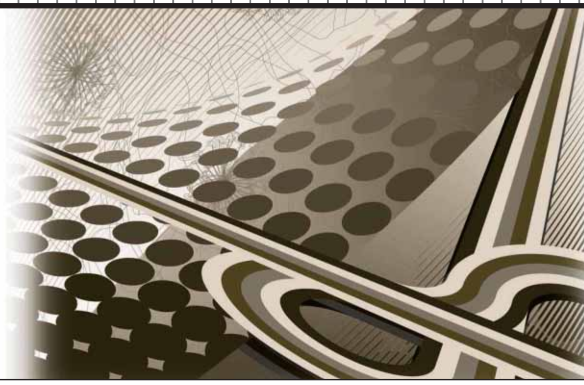
Design & Test's July/August issue features seven theme articles on FPGA accelerator research. Guest editors George A. Constantinides of Imperial College London and Nicola Nicolici of McMaster University see the field as approaching a threshold that could expand its applications beyond high-data-rate digital signal processing to include high-performance scientific computing. "It is our hope," they write, "that bringing together the research in this special issue will help readers bridge the historically distinct FPGA, high-performance computing, and numerical analysis communities."

IEEE Annals of the History of Computing

"Casinos and the Digitization of the Slot Machine, 1950-1989" is one of *Annals*' collection of articles from new voices on new topics for its April-June issue. Author Cristina Turdean of the University of South Carolina looks at the slot machine's transition from a crude mechanical device in the 1950s to a sophisticated digital gambling device by the early 1990s. "Suspicion and fraud, the gambling environment's two main descriptors," she writes, "guided the slot machine development toward protecting the game outcome and the machine's cash."



The Chicken Bus



David Alan Grier, *George Washington University*

As with all engineering problems, we need to balance the twin factors of stability and control and do so in a way that doesn't damage either of them.

On closer examination, the car wasn't as distinctive as it appeared. To identify it as a Chicken Bus or, more accurately, a Chicken Sports Car, is a disservice to the Central American mechanics who've learned how to adapt old American school buses to serve as a transportation system that can carry people across the countryside and deliver chickens to the city, goats to the slaughterhouse, or cash crops to market. To advertise their special role in the local transportation system, these vehicles are usually painted with gaudy colors and clever decorations and are called Chicken Buses.

The vehicle that was circling the block had the paint job of a Chicken Bus but the attitude of something very different. It was a low Italian sports car of recent vintage, perhaps a Lamborghini, and it had been decorated in a dark and brooding checkerboard scheme that included arbitrary accents of color and enticing portraits of beautiful women and strong men.

We heard the car long before we saw it. The four of us were slowly walking along the street on our way to dinner when we heard sounds that we interpreted as coming from a badly rusted exhaust system. As we looked for the source of the noise, we saw the Chicken Lamborghini accelerate for about 10 or 15 yards, accompanied by a loud roar. As the car got close to a line of stopped traffic, the driver hit the brakes, which produced a plume of smoke from the rear tires and three or four loud belches from the tailpipe.

We watched the car circle the block three times, each with the same dramatic effects. The third time, the driver was caught at a light and decided to rev his engine, perhaps just to prove that he could.

"Poor guy," said one in our group. "He's got the itch, and he's got it bad."

"At least $250,000 bad," said another, who seemed to know the price of handmade automobiles.

"And nobody's paying attention to him," added a third.

"Certainly not the kind of anybody that he hopes is going to be impressed," returned the first. "Just because you can make a loud noise doesn't mean you have anything to say."

DIFFERENT TYPES OF INNOVATIONS

The Chicken Car quickly faded from the conversation as we sat down to dinner. We had been attending a technical meeting all day and had more interesting things to discuss. In particular, I could report that I had been sitting next to a young engineer who appeared to be using a radical version of a popular tablet computer, which had an operating system unlike anything I'd ever seen. The interface involved spinning globes and twisted ribbons. It had a file system that seemed quite strange to me but was completely obvious to the owner. As the machine responded to commands, the screen would unleash various animated objects that seemed to represent data or applications, or both, or neither.

At a break, I asked the young man about the machine, and he confirmed that it was indeed a recent model tablet that was running an operating system of his own design. Something in my voice must have expressed shock or disbelief, for he immediately described the system's origins. It was based on code developed at a university in southeast Asia, perhaps Kuala Lumpur. He had removed the unnecessary sections, written a new kernel and a few drivers, merged some key modules from an underutilized American operating system, and restructured the entire thing.

"It's lean," he said, "and fast. It has none of the stovepipes that you find on commercial operating systems."

I asked if it could run any of the conventional tablet applications.

"No," he confessed, but he noted that he'd been able to import all the software he needed.

Few of us can assemble a modern operating system, even if we're borrowing parts from other sources.

Published by the IEEE Computer Society

Yet, like the mobile work of art that circled the block as we walked to dinner, this machine seemed to be entirely a personal accomplishment. It demonstrated the depth of his skills, but it had little impact on a bigger community. For the moment, the opportunities for introducing a major new tablet operating system seem to be limited. Those who guide new ideas toward widespread application are looking in other places for innovation.

Both the Chicken Car and the Chicken Tablet, if we can call it that, represent types of innovation. The first is a conventional concept in a rare and expensive container. It speaks to the need of young men to make a mark for themselves and get attention from others. The second is an unusual concept in a more common shell. It may represent a true novelty, but it also may be so personal that it isn't possible to label it as a valid form of innovation.

The nature of innovation is an issue that's of great concern to many of my colleagues. By innovation, they mean the ability to create new goods or services that produce wealth and thereby increase accumulated capital. They're concerned that we aren't generating enough innovation, that it's happening in the wrong part of the world, or that it's failing because of a lack of capital, proper education, moral fiber, or a good, filling breakfast.

Most important, they're concerned that we aren't seeing enough innovation because the wrong people are in control and wrong ideas are holding sway. The "future unfolding right now is very different from past innovation," wrote one critic of current innovation strategies. This writer claimed that the last two decades of the 20th century were a golden age in which open computer architectures and open networks encouraged inventors to develop new products and services. "The future is not one of generative PCs attached to a generative network," he warned. "It is instead one of sterile appliances tethered to a network of control."

BUILDING BLACK BOXES

Certainly, the technology industry flourished in an environment with open architecture machines that allowed individuals with little capital—and sometimes limited training—to develop innovations. Yet simultaneously, that industry has advanced by an opposite process, one that closes technologies and connects them to existing bodies of knowledge. This process locks ideas into black boxes, which have interior mechanisms that can't be altered. "Science has two faces," wrote philosopher Bruno Latour, "one that knows and the other that does not know yet." Both of these elements are needed for technology to advance.

Codd was able to extract the basic concepts of data manipulation from its surrounding environment and place it in a framework that many found convenient.

Almost any technical innovation closes an open system, limiting access to ideas that were once easily modified. A central example within the field of computing technology is E.F. Codd's development of the relational database. In 1970, Codd published a paper that not only established the fundamental ideas for databases but also illustrated the pattern by which we define a black box for technology. In his paper, Codd described a basic set of ideas that became central to database design. "We take for granted the benefits brought to us by relational databases," explains a recent college textbook.

Strictly speaking, the process of building a black box shouldn't be confused with the process of invention, the act of being the first to identify and describe an idea. Over the course of my career, I've received many e-mails from computer scientists who feel that they didn't receive proper credit for ideas they discovered or invented. While I sympathize with these researchers and always try to find an interesting story that values their work, I have to remember that our society honors those who provide the most utility from an idea. The first across the line aren't necessarily those who bring the greatest value.

In building a black box, a researcher is creating a self-contained system that can operate without exposing its inner workings. Hence, most of the ideas that form the black box will already exist. The ideas that provided the basis for relational databases had been discussed for almost a decade before Codd wrote his seminal paper. Indeed, in 1966, workers at RAND created a database system that had most if not all of the features of a relational database.

When he defined the concepts of relational databases, Codd was far from the most distinguished researcher in the field. Most of his writings dealt with problems of timesharing. He'd only been studying databases for a few years. Yet, he was able to extract the basic concepts of data manipulation from its surrounding environment and place it in a framework that many found convenient. The "problems treated here are those of data independence," he wrote. "[This paper] provides a means of describing data with its natural structure only—that is, without superimposing any additional structure for machine representation purposes."

Codd's work required about 10 years to establish itself as a unified technology, as a set of ideas that could be used without dealing with their internal operations. The first recognition came in the academic literature. His paper had 59 citations by 1975 and another 460 by the end of the decade. In parallel with the approval of researchers came interest from industry. By 1980, a half-dozen firms were offering relational database software, including two start-up companies that would have major roles in the software industry.

No matter how quickly ideas are accepted by researchers or form the basis for an industry, they won't survive without a careful and thorough defense. There are always others who think they have a better idea, a more productive plan, a greater stake in an alternative approach. For the next 20 years, Codd defended his ideas as the best ways to create and manage data. In making this defense, Codd was backed by the resources of his employer, IBM, which had made substantial investments in relational databases. With these resources, Codd focused on the technical value of his database. "After the publication of my papers," he explained, "numerous articles began to appear proposing new approaches to database management." He claimed that most of these articles attacked his model with "the false claim that the relational model contains no features for representing the meaning of the data." He would then dismiss these claims and argue that any new contribution could be incorporated into his model.

GENERATIVE INNOVATION

For the past five or six years, policymakers have talked about the need to create a system of innovation that they describe as generative. In using this term, they're usually referring to the "capacity to produce unanticipated change through unfiltered contributions from broad and varied audiences." They debate various strategies of creating such a system and argue about whether it's best done in open source environments, under the control of private capital, or under the watchful eye of a central government.

While the watchful central government has perhaps been the least effective shepherd of innovation, it has still played a role in the process of closing and opening technology that has spurred new ideas. At key moments, we can declare one technology closed and stable so that researchers can use it as a tool for future development. The forces that close and stabilize a technology can be government regulation, the clout of capital, or the mutual agreement of wary parties who believe that a stable market is the best place for them to develop new ideas. Even in the most congenial market, they'll have to do battle with the army of invisible hands that guides prices to a point that balances supply and demand, evaluates innovation, and leads some to success while it crushes the rest beneath its heavy blow.

My colleagues have also been concerned that the global systems of innovation were failing to inspire truly original ideas because too many technologies were in the hands of too few institutions. After generations of producing wonderful ideas, we were reduced to a situation in which we could do little with those ideas. The owners might let us borrow an idea for a time, give it a clever coat of paint, and drive it around the block in the vain hope that we might encounter someone or some idea of interest. Just because you can combine old ideas doesn't mean that any of those combinations will be novel.

As with all engineering problems, we need to balance the twin factors of stability and control and do so in a way that doesn't damage either of them. We must provide standard concepts and ideas so that anyone can build a new database, a new processor, or a new car. We also need to support a venue that will emphasize the advantages of the new ideas. It may look like a Chicken Bus to a group of critical guys standing on a street corner, but someone may be intrigued to give it a spin and see where it will go. C

David Alan Grier is an associate professor of international science and technology policy at George Washington University and a senior member of the IEEE Computer Society. Contact him at grier@computer.org.

32 & 16 YEARS AGO

AUGUST 1979

SOFTWARE ASSURANCE (p. 7) "Although long recognized as a serious problem, many of the issues encountered in assuring the quality of computer programs have only received the level of attention they deserve in the last decade."

"The six papers included in this issue form a cross-section of software quality assurance methods that ranges from detailed design of critical systems, to establishment of IEEE Computer Society proposed standards for software quality assurance procedures."

REACTOR PROTECTION (p. 10) "Computers in the nuclear field have been used mainly as passive components—i.e., for data acquisition and data representation on display and logging devices. However, in a reactor protection system, the computer may now play a more active role.

"The computerized reactor protection system considered here must scan about 200 measuring points every second. …"

CERTIFICATION TESTING (p. 20) "Several past studies have shown that the various testing phases can account for up to 50 percent of the total resources spent for the development of a software system. …"

"In spite of the large investment in resources, little systematic attention has been paid to this portion of the development life cycle. … However, on further consideration, it becomes apparent that reliability cannot be tested into a program. The role of testing is to locate errors in what is hoped to be a well-designed system."

ERROR DETECTION (p. 26) "Two software testing techniques—static analysis and dynamic path (branch) testing—are receiving a great deal of attention in the world of software engineering these days. However, empirical evidence of their ability to detect errors is very limited, as is data concerning the resource investment their use requires. … this paper seeks (1) to demonstrate empirically the types of errors one can expect to uncover and (2) to measure the engineering and computer time which may be required by the two testing techniques for each class of errors during system-level testing."

AUTOMATED TESTING (p. 33) "One of the more popular tools for supporting software testing is the automated testing analyzer, or ATA. … A number of such tools are presently in use in a variety of applications, and perform functions such as static analysis, assertion processing, and test data generation. Most provide a testing 'coverage' based on the effect a set of tests has on the internal control flow of the program under test."

STANDARDS AND METRICS (p. 37) "Software quality testing as used in the following discussion includes activities which determine the readiness of the software for formal acceptance testing. … These activities are generally supportive of formal testing, but do not focus primarily on the validation of functional performance requirements. In some cases the same criteria used in quality testing can also be used in acceptance testing. The term software includes the computer program, the data base, and associated operation and maintenance documentation."

ASSURANCE PLANNING (p. 44) "… There are many software engineering methodologies by which software is developed, and no standard methodology exists to ensure that software acquires the required characteristics as development progresses. Moreover, there is no industry standard for expressing plans to implement assurance programs for software development."

MICROPROCESSOR MAINFRAMES (p. 68) "Achieving mainframe functionality with LSI technology is of great interest to computer designers because of the anticipated dramatic improvement in cost-performance. One design approach that is often proposed is to interconnect a set of microprocessors so that each microprocessor executes different user jobs. The total throughput of this concurrently executing set of microprocessors is expected to be equivalent to the throughput of a single mainframe processor. …"

SOCIAL SOFTWARE (p. 84) "The development of software takes place in an environment of social relationships. For example, the preparation of requirements is built upon the intricate relationships between the organization that needs a system and the organization that is to program it. The management of software development is itself a social activity. The so-called maintenance of operational software is often in actuality the enhancement of the original product; these enhancements again arise out of a complex interface to the using organization."

HANDHELD COMPUTERS (p. 99) "Plans to produce a line of practical handheld personal computers have been announced by Matsushita Electric Industrial Co., Ltd., Osaka, Japan. Discussing his company' s agreement with California-based Friends/Amis Inc., Dr. Shunkichi Kisaka, senior managing director in charge of research and development at Matsushita, said that the joint efforts of Matsushita and Friends/Amis' will give birth to an entirely new field of consumer electronics.' "

32 & 16 YEARS AGO

AUGUST 1995

WIN95 (p. 9) "… Apple' s fascination with GUI design is distracting the company from its number one objective: gaining market share. Only a radically superior OS, or uncommon marketing, can save Apple. Once a technology leader, Apple is on the defensive: it must play catch-up, or die. Meanwhile, back in Cupertino, someone is redrawing the org charts. Two thousand years ago, someone in a similar situation played a fiddle."

NETWORK STANDARDS (p. 12) "Users wishing to upgrade from 10BaseT Ethernet to a 100-Mbps LAN have two high-speed alternatives from which to choose. After months of vigorous debate, the IEEE Standards Committee has formally approved the Fast Ethernet Alliance's 100BaseT Fast Ethernet standard and the Hewlett-Packard-supported 100VG-AnyLAN standard. The two standards, although based on decidedly different technologies, are being tested in the user community."

SOFTWARE SUSTENANCE (p. 44) "Program understanding is a major factor in providing effective software maintenance and enabling successful evolution of computer systems. For years, researchers have tried to understand how programmers comprehend programs during software maintenance and evolution. Five types of tasks are commonly associated with software maintenance and evolution: adaptive, perfective, and corrective maintenance; reuse; and code leverage. …"

COMPUTING IN MEXICO (p. 56) "The North American Free Trade Agreement (NAFTA) brings Canada, Mexico, and the US closer as commercial partners and raises many questions because of the disparity in the size and strength of their economies. In the long run, the low labor costs said to give Mexico a competitive edge will become less relevant, since Mexican government officials see NAFTA as a tool to create jobs, increase salaries, and raise average family income.

"The real issue is whether Mexico can effectively compete in a larger market and stimulate economic growth. To this end, technological development is one of the most important factors. …"

COMPUTING IN CHINA (p. 64) "… During the 1970s and early 1980s China tried to develop a full line of computer hardware and software incorporating indigenous technology. In the mid-1980s the government began emphasizing IT use throughout the economy, development of production capability by joint ventures with foreign multinationals, and production of low-end IT products for export.

"This more pragmatic IT policy has led to a dramatic increase in IT use, a rapidly growing computer industry, and expanding exports. …"

LINUX (p. 74) "Linux has been under development for only a few years. As is to be expected, the initial versions were unstable and of interest only to die-hard hackers. However, Linux has now matured and can be used for serious work. Many of the powerful compilers and utilities developed by the Free Software Foundation under its GNU (GNU is Not Unix) project can run on Linux. The X Window System from MIT, the de facto standard for windowing under Unix, is also available for Linux in an Intel-386/486 version called XFree86."

FIBRE CHANNEL (p. 88) "… in 1988 ANSI committee X3Tll initiated development of Fibre Channel, a switched protocol capable of transmitting at rates exceeding one gigabit per second, while still supporting existing protocols over both optical fiber and copper cables. Fibre Channel combines the best attributes of legacy channels and networks into a single standard that is a generic transport mechanism for data, voice, and video. It is the key to scientific and business applications implemented in open and distributed architectures, because it removes the barriers to performance presented by the old methods of data communications."

SOFTWARE PATENTS (p. 99) "… In its recently published guidelines …, the US Patent and Trademark Office (PTO) said computer software programs stored in a tangible medium, such as a floppy disk, are patentable and must be examined to determine whether the substance of a computer-program-related invention is a significant advance over prior technical achievement justifying the grant of a patent. In the past, the PTO had simply refused to examine the substance of such an invention."

SOFTWARE ENGINEERING (p. 100) "One of the factors associated with recognized formal professions, … is a well-defined body of knowledge. Often this knowledge includes many subsets of more specialized knowledge. Also necessary is an appropriate academic curriculum to transfer knowledge to students.

"In keeping with the above principles, the Joint Task Force on Software Engineering Ethics and Professional Practices, established by the IEEE Computer Society and the ACM, is working to have software engineering recognized as the 37th engineering profession."

PDFs of the articles and departments from Computer*'s August 1979 and 1995 issues are available through the IEEE Computer Society' s website: www.computer.org/computer.*

Editor: Neville Holmes; neville.holmes@utas.edu.au

Ten Things to Know When Your Patent Application Is Pending

Brian M. Gaff, *Edwards Angell Palmer & Dodge LLP*
Catherine J. Toppin, *General Electric Corp.*

The second in a series of three articles provides basic points to keep in mind when filing for a patent.

In the first article in this three-part series, we listed 10 "do's and don'ts" to keep in mind while you consider whether to file a patent application on your invention. Assuming that you've decided to file, there are other issues you should think about as you begin the application process and during the two to three years typically needed to get a patent.

APPLICATION PROCESS

After you file with the US Patent Office (USPTO), your application goes through a series of administrative steps: it is assigned a serial number (which is not the same as a patent number), checked for completeness, and handed out to a patent examiner for review. Because of the number of applications in the queue, it may take approximately one to two years before the patent examiner can begin reviewing your application.

The process of reviewing the application is called the "prosecution" phase. During this phase, the patent examiner will send out Office Action notices that "allow" or "reject" the claims in the application. To keep your application alive, you must respond to these Office Actions with precise legal arguments and within certain deadlines. Once begun, prosecution typically takes one to two years.

TEN CONSIDERATIONS WHILE YOUR PATENT IS PENDING

When filing and during prosecution, keep the following 10 important points in mind:

Choosing a patent attorney

Unless you have significant experience applying for and prosecuting patent applications, you should hire a patent attorney to handle your application. Mistakes made during prosecution can be fatal to your application. It's important that you choose a patent attorney who has a background in the subject matter of your application.

Funding your vision

The cost of prosecution can be significant, particularly for complex inventions and situations in which the prosecution can last longer than the usual amount of time. For example, analyzing and responding to an Office Action in a moderately complex case can cost $5,000 to $7,000, so it's important to ensure that your budget can support these expenses.

International and foreign protection deadlines

You should apply for non-US patents if you plan to market your invention outside the US or if you expect competition outside the US. Take care, however, to note the varying deadlines for applying for non-US patents at the time you file your US application.

Patent term adjustment

Delays during prosecution are commonplace. Delays that the USPTO causes are usually credited to the applicant in the form of extra time added to the life, or "term," of a patent. However, delays that the applicant causes can subtract from the term, so it's important to be responsive to communications from the USPTO and avoid asking for extensions of time.

Reviewing claims during prosecution

The claims in your application are typically amended during prosecution to address the examiner's concerns about their patentability. Ensure that

0018-9162/11/$26.00 © 2011 IEEE

any amendments you're considering won't result in your claims failing to cover the products that embody your invention.

Presenting your case

During prosecution, you will have opportunities to have an interview with the examiner. Interviews may be in person at the USPTO or, more typically, conducted over the telephone. It's usually helpful to have an interview to discuss the examiner's concerns about your claims and potential amendments to overcome those concerns. Interviews help avoid misunderstandings and can facilitate prosecution.

Continuing prosecution with RCEs

There's a point in prosecution when you're no longer permitted to make substantive amendments to your claims. If by then you haven't convinced the examiner to withdraw the rejections of your claims, then you must decide whether to abandon your application or file a Request for Continued Examination (RCE). Choosing this option and paying the corresponding $810 official fee will give you another opportunity to amend your claims and convince the examiner to allow them.

When to provide a declaration

Some claim rejections can be overcome by showing the examiner that your invention led to unexpected results or commercial success. Arguments based on these factors can be complex and must be presented in a declaration—a sworn statement—that sets forth specific facts and reasoning. Be aware that the cost of prosecution can increase if you choose to raise arguments using this type of declaration.

When to appeal the examiner's decision

You may reach a point during prosecution when it's apparent that you and the examiner will never agree on the patentability of one or more of your claims. Instead of abandoning your application, consider filing an appeal. Appeals in prosecution go to a board of specialized patent judges in the USPTO. You're allowed to argue your case before the board, and the board has the authority to overrule the examiner. Costs for an appeal can be significant, so ensure that you plan for it in your budget.

Covering new discoveries with a CIP

Inventors who continue their research during the prosecution phase commonly make new discoveries that are related to the subject matter of their pending application. One way to apply for patent protection on any new discoveries is to file a Continuation in Part (CIP) application. A CIP is designed to capture new but related inventive features. It's a separate application that can lead to a separate patent. The costs associated with filing a CIP are usually less than the costs of the original application because much of the legwork of the original application can be reused in the CIP.

Keeping these 10 things in mind and—most importantly—staying engaged during the lengthy prosecution phase will help you get one or more patents that give your invention the protection it deserves.

As you head toward the goal of obtaining a patent, there are many alternatives and varying paths that you can follow. You should evaluate the costs and benefits of each as well as the overall impact that your choices could have on the likelihood of your application successfully becoming a patent. **C**

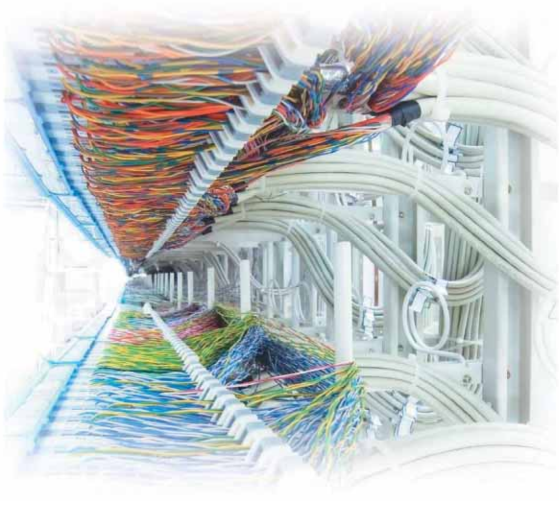
Brian M. Gaff, a partner at the Edwards Angell Palmer & Dodge LLP law firm, is a senior member of IEEE. Contact him at bgaff@eapdlaw.com.

Catherine J. Toppin, a patent counsel at the Global Patent Operation of General Electric Corp., is a member of IEEE Contact her at catherine.toppin@ge.com.

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

TECHNOLOGY NEWS

OpenFlow: The Next Generation of the Network?



Steven J. Vaughan-Nichols

Software-defined-network technologies like OpenFlow could change how datacenters, cloud systems, and perhaps even the Internet handle tomorrow's heavy network loads.

New technologies have changed and complicated the nature of networking. For example, the demands of cloud computing and massive datacenters have made effective networking much more complex.

To cope with these demands, operators want their networks to be smarter and they want to be able to better control and manage them. In response, interest has increased in *software-defined networks*.

SDNs run the network control plane in software, generally on standard servers operated separately from network devices such as switches. This gives network administrators more fine-grained control over traffic flows.

The idea for SDNs has been around for about a decade. Recent implementations by companies such as Cisco Systems and Juniper Networks have been proprietary and thus haven't worked with equipment from different networking vendors.

However, vendors and network operators are beginning to look closely at OpenFlow technology, which promises to bring interoperability and better performance to SDNs. With OpenFlow controllers, administrators could define flows and determine how packets are prioritized and forwarded through switches over a local- or wide-area network (LAN or WAN).

In essence, OpenFlow transfers control of network traffic flows from the infrastructure—the switches and routers—to administrators.

Proponents say the technology will yield flexible, secure networks that have fewer traffic issues and are less costly to construct and run. One vendor has already released an OpenFlow product line. Others have released test samples of OpenFlow equipment.

In March 2011, companies such as Cisco Systems, Facebook, Google, and Microsoft formed the Open Networking Foundation (ONF) to promote OpenFlow technology and the OpenFlow Switching Protocol.

Nonetheless, some industry observers speculate that OpenFlow won't scale well and that companies could add proprietary extensions to the technology, eliminating the benefits of interoperability.

PROBLEMS TO SOLVE

The rapid growth in cloud computing is driving the need for flexible, reliable, secure, and well-managed network backbones, noted Bruce Guptill, head of research for Saugatuck Technology, a market analysis firm.

This requires more intelligent and efficient management systems to coordinate thousands of interlinked routers and switches, he explained. Currently, routers and switches offer only limited user programmability.

Administrators need detailed and scalable control to maximize datacenters' effectiveness, noted University of California, Berkeley, professor Scott Shenker, an ONF board member and OpenFlow researcher.

While SDNs promise to help with these issues, vendors—such as Cisco and Juniper—provide only proprietary implementations of the technology.

Each vendor has its own API and SDN functionality, which can limit the ability to engineer and manage traffic across equipment from multiple vendors. Also, a traditional switch handles both the fast packet

0018-9162/11/$26.00 © 2011 IEEE

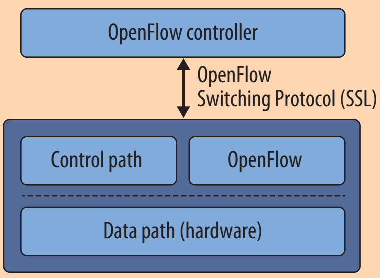


Figure 1. In an OpenFlow network, the OpenFlow controller handles high-level routing decisions instead of the local switch, as is typically the case. The switch's processing power can then be used for faster packet forwarding and other tasks. Administrators can work with the controllers to more effectively run their networks.

forwarding (data path) and the high-level routing (control path).

OpenFlow makes the control function independent of the hardware it controls. This speeds up the forwarding and routing processes.

Also, noted ONF president Urs Hoelzle, Google's senior vice president of engineering, "In an OpenFlow network, all the intelligence will be in a central [server], so it's easier to do complex things."

OPENFLOW

Researchers at Stanford University and UC Berkeley began working together on SDNs in 2002 and on OpenFlow in 2008. Vendors such as Juniper began releasing proprietary OpenFlow-related SDN products in 2010.

"OpenFlow is the first viable [SDN] approach," said Alex Reimers, a software engineer for Big Switch Networks, an OpenFlow switch vendor.

How it works

Most modern Ethernet switches include flow tables that describe how to move a packet efficiently from sender to destination. Each vendor's flow table is different, but researchers identified a set of functions—such as quality of service and traffic reporting—common to most high-end switches.

OpenFlow standardizes this common set of features.

Separating data path and control path. OpenFlow separates a switch's data path and control path activities, as Figure 1 shows.

The data-path functionality still resides on the switch, but a controller, typically a standard server, handles high-level routing decisions. The switch and controller communicate via the OpenFlow Switching Protocol.

The controller can, for example, tell switches to impose policies—such as sending data by routes that are the fastest or have the fewest number of hops—on network traffic flows, explained Stanford professor and OpenFlow designer Nick McKeown.

Interface. OpenFlow provides a common API that lets administrators program what they want to happen in the network, as well as how to route packets, perform load balancing, and handle access control, explained UC Berkeley's Shenker.

"There are two basic components: a software interface for controlling how packets are forwarded through network switches and a set of global interfaces upon which more advanced management tools can be built," added the ONF's Hoelzle.

Benefits

OpenFlow offers several important benefits.

Performance and cost. By removing the control-processing load from the switches, OpenFlow lets the switches focus on moving traffic as fast as possible.

By virtualizing network management, OpenFlow enables networks that are less costly to construct and run.

Implementing and testing new features. OpenFlow lets administrators deploy new features using the existing network architecture by adding them in OpenFlow software.

The features would then work on multiple platforms. Users wouldn't have to implement them in each switch vendor's hardware or firmware.

Via its open API, OpenFlow also enables administrators and researchers to write their own control software and try out important new switch functionality at full line rates, noted Big Switch cofounder and CEO Guido Appenzeller, one of OpenFlow's creators.

This has been difficult in the past because major vendors' routers and switches lacked common APIs.

Security and management. OpenFlow's centralized controller gives administrators a unified view of the network, which enables better management, security, and other capabilities.

By letting administrators see the overall traffic flow more clearly, Hoelzle said, OpenFlow makes spotting intrusions and other problems easier.

OpenFlow also lets administrators prioritize different types of traffic and develop policies for how the network handles congestion and equipment problems, he noted.

The technology also promises to let administrators create virtual network topologies, setting up virtual LANs (VLANs) or WANs (VWANs) as needed without physically changing the network.

In the process, administrators could set up a centralized virtual control plane, facilitating network management. This could be especially valuable for datacenter management. For example, a network manager could easily set up a VLAN for a new customer from an OpenFlow controller instead of having to work with individual switches or individual vendors' switches.

Cloud computing. In cloud computing, data and applications are on network-based computers. For this approach to function optimally, OpenFlow proponents say smarter networks are necessary to, for example, orchestrate the behavior of the many switches involved in the process.

Products and implementations

NEC Corp. of America's ProgrammableFlow products—including controller-based software, switches, and a traffic-monitoring system—are the only OpenFlow-compatible items on the market.

Brocade has announced it will offer the technology in switches later this year. Vendors such as IBM and Juniper are shipping OpenFlow equipment as test samples.

OpenFlow software is also available on various Linux distributions, including CentOS, Fedora, and Ubuntu. This lets a Linux machine act as an OpenFlow switch or control server.

OpenFlow testbed networks are already available in some academic and research settings.

The Internet2 consortium, Indiana University, and Stanford's Clean Slate research program—as part of their joint Network Development and Deployment Initiative, designed to create a new network platform to support scientific research—are building a new nationwide OpenFlow network. This would be the first large-scale OpenFlow network available for research or production use.

CONCERNS AND CAUTIONS

Because OpenFlow centralizes network-control functions into a single server, Force 10 Networks chief marketing officer Arpit Joshipura said, there are concerns the technology might not scale.

Currently, all switches in a network help with management. As a network grows, the additional switches help the system scale.

Some industry observers are worried about security. They fear that because OpenFlow places all management functionality in one virtual server, hacking the network will be easier.

"OpenFlow will fit where you need less security," said Bill Seifert, chief technology officer of Avaya, a business communications vendor.

Some say that while basic OpenFlow might work well, vendors could add proprietary extensions to the technology and make their versions incompatible with other implementations.

FLOWING FORWARD

The ONF's Hoelzle said OpenFlow "will allow networks to evolve and improve more quickly than they can today."

"The value proposition of OpenFlow is that it reduces the friction of implementing network changes for many organizations," said Greg Ferro, a UK-based freelance network architect, engineer, and designer. "In public cloud networks, where low cost is a primary driver, it's likely they will adopt OpenFlow as the primary configuration tool for almost all network requirements."

If OpenFlow works as promised and lets users get more productivity from their network equipment than they do today, the technology could be popular with customers and profitable for vendors, added Kyle Forster, Big Switch's cofounder and vice president for sales and marketing.

Jon Oltsik, an analyst for Enterprise Strategy Group, a market research firm, was less optimistic. "OpenFlow can still be classified as more science project than real-world implementation," he said. "Yes, it could change the dynamics of networking but most users want switches and routers that just work. If OpenFlow can get to this level of functionality, simplify network architectures, and streamline network operations, it may [succeed]."

However, he added, if OpenFlow demands new skills and requires complex network programming, it may not become popular. ■

Steven J. Vaughan-Nichols *is a freelance technology writer based in Mills River, North Carolina. Contact him at sjvn@vna1.com.*

Editor: Lee Garber, *Computer*; l.garber@computer.org

Experts Say Rapidly Spreading Botnet Will Be Tough to Stop

A security vendor says the current version of a rootkit that has infected 4.52 million computers and formed them into a massive botnet in just three months this year is highly sophisticated and will be difficult to defeat.

Kaspersky Lab said the TDL-4 rootkit's operators use the botnet to plant adware, keystroke loggers, additional malware, and other problematic software on PCs.

The operators earn money by renting their botnet to other hackers for purposes such as distributed denial-of-service, spam, and phishing attacks. Researchers say customers who want to anonymize their communications can pay hackers to use one of the botnet's machines as an Internet traffic proxy.

TDL-4 spreads via Trojans, which affect users with vulnerable systems who visit infected pornography and file-sharing sites.

Researchers recently found that the rootkit also creates a Dynamic Host Configuration Protocol server that directs users to a malicious DNS server, which then redirects them to infected webpages.

TDL-4 is the fourth version of a rootkit that debuted in 2008 and has also been called Alureon and TDSS. Security experts say the rootkit has always been well designed and that its operators have added dangerous new features to its current iteration.

For example, TDL-4 uses an advanced custom encryption-scheme that keeps victims' network-monitoring tools from intercepting and deciphering commands and other transmissions between the hackers' control servers and infected machines.

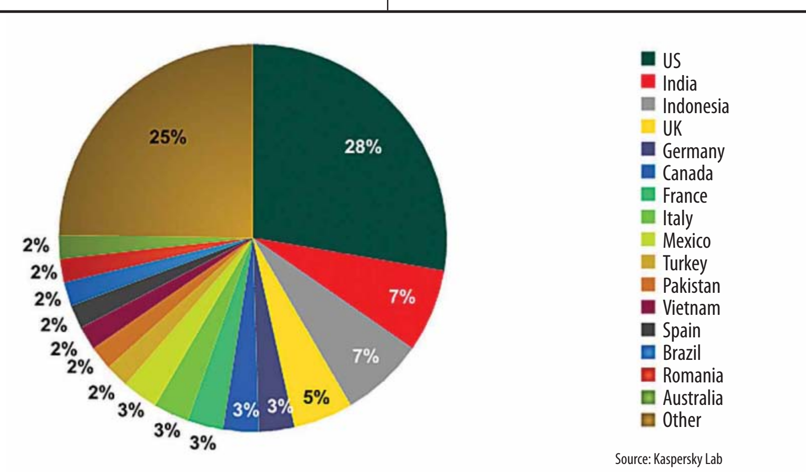
The rootkit can transmit commands and updates via a private communications system, as most botnets do. Law enforcement officials frequently try to fight botnets by taking down their private command-and-control systems. TDL-4 combats this by also being able to employ the public Kad peer-to-peer file-sharing network.

In its attacks, the rootkit infects a PC's master boot record, which lets it run before the OS boots up and anti-malware software kicks in.

TDL-4 includes features that delete 20 other types of malware from infected computers and blacklist the addresses of the command-and-control servers these other malicious programs use. Disabling these programs reduces

the chance that security systems will scan infected machines and possibly find TDL-4.

Kaspersky researchers have found a flaw in the rootkit's code. They used one of these flaws to spy on its databases, which they hope will help their investigation.



Research by security vendor Kaspersky Lab shows that the TDL-4 rootkit has been found in computers in many countries, with 28 percent of the infections in the US. TDL-4 infected 4.52 million computers and formed them into a massive botnet in just the first three months of this year.

ICANN Increases the Number of Top-Level Domains

The Internet Corporation for Assigned Names and Numbers has approved a major expansion of generic top-level domains in a step that marks a big change in Internet operations.

During recent meetings, ICANN's board of directors voted to let generic top-level domains—the letters following the last dot in a Web address, such as.com, .net, and .org—to end in almost any word in any language. Previously, there were only 22 gTLDs.

The new plan would, for example, let companies and other organizations turn their names into top-level domains. In addition, said ICANN president and CEO Rod Beckstrom, "[The] decision respects the rights of groups to create new top-level domains in any language or script."

The organization says it expanded gTLDs to increase choice, innovation, and marketing opportunities.

ICANN plans to accept applications for new gTLDs during an initial period from 8 January to 12 April 2012 and to charge about $185,000 for evaluating each proposed domain's application. There will be additional application periods in the future.

When more than one organization applies for the same gTLD, ICANN will use an established process to select the recipient. ICANN also has processes that let third parties object

Published by the IEEE Computer Society

to a top-level domain because, for example, a gTLD represents another organization's trademarked name.

In the near future, ICANN says, it will start a campaign to educate people about the changes and opportunities the new system offers.

The topic of gTLDs has often been controversial. For example, ICANN and several national governments have argued over how much influence governments and trademark owners should have over the domain-name-creation process.

Eight gTLDs—.com, .net, .org, .edu, .gov, .int, .mil, and .arpa—existed before ICANN's creation in 1998. The organization created seven more in 2000 and another seven in 2004.

ICANN is responsible for various Internet-governance tasks, such as managing IP addresses, assigning address blocks to the five regional Internet registries—like the American Registry for Internet Numbers—and managing TLDs.

USING TECHNOLOGY TO HELP THE THIRSTY

People living in developing countries in Asia, Africa, and Latin America face great uncertainty over water supplies.

Many communities have water available through pipes, but the unreliable supplies enable service for only a few hours at a time. This is a particular burden because there is no schedule for when utilities will distribute the vital liquid. Residents must wait, often days at a time, to make sure they are home when water is available.

NextDrop—a project of the University of California, Berkeley's School of Information, Department of Civil and Environmental Engineering, and Goldman School of Public Policy—has developed a messaging-based approach to solve this problem.

The approach starts with water-utility workers calling NextDrop's interactive voice response subsystem when they open local and feeder valves.

The subsystem then sends text messages to local residents, many of whom have cellular phones, up to an hour before water delivery. Subsequently, NextDrop randomly contacts recipients to determine the accuracy of the information that the utility sent. In case of inaccuracy, the system notifies the utility to address the problem.

NextDrop provides data on delivery outcomes for water-utility engineers. The engineers and residents can view delivery status in real time via a dashboard.

So far, the NextDrop organization is operating a pilot project with Hubli, India's water utility. Researchers hope to serve 1,000 families by March 2012 and the whole city, which has a population of about 800,000, by July 2013.

NextDrop has offered its services for free so far but plans to begin charging in the near future.

NextDrop started two years ago as an entry in a contest held in a UC Berkeley civil-engineering class. The project won the competition.

It has also won numerous awards from organizations such as the Clinton Global Initiative University, Center for Information Technology Research in the Interest of Society, the Knight News Challenge, the Global Social Venture Competition, and the Global Social Entrepreneurship Competition.

MIT Microchips Could Revolutionize Healthcare

MIT researchers are developing energy-efficient microchips that could operate within wearable or implantable devices that monitor medical patients for health problems.

Scientists in the university's Microsystems Technology Laboratory (MTL) are working on tiny chips for biomedical devices that could help diagnose or monitor multiple medical issues, including heart problems.

This approach could radically change the nature and cost of healthcare, according to MIT visiting scientist Dennis Buss.

"Microelectronics have the potential to reduce the cost of health care in the same way they reduced the costs of computing in the 1980s and communications in the 1990s," said Buss.

A key to the MIT approach is developing low-power chips to run biomedical monitoring systems and handle their communications.

The MTL researchers are currently working on an electrocardiogram system and hope to develop one that can measure other vital signs, including breathing rate, blood pressure and oxygenation, pulse, and body temperature.

Patients could wear a device during their everyday activities, rather than just when they're in medical offices, and still provide the ongoing vital-sign measurements critical for diagnoses and other purposes, said physician and MIT associate professor Collin Stultz.

The prototype MTL heart monitor is L-shaped and 4 inches long on each side. It sticks to the chest and has no external wiring. The device stores two weeks of data in flash memory and uses just two milliwatts of electricity.

Stultz and his colleagues designed an algorithm that uses data the device gathers to determine a patient's risk of death. Currently, this analysis occurs only after data is downloaded from the monitor. The researchers are working on ways to incorporate their algorithm into the chip, which would enable an automatic notification of impending problems.

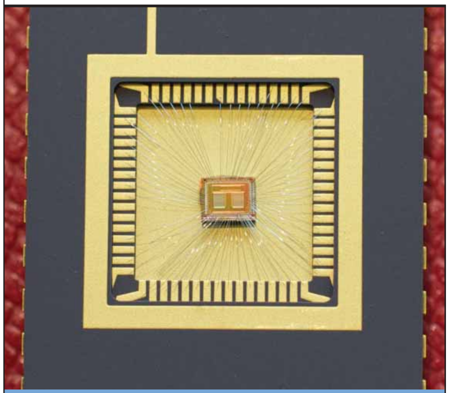
The MTL scientists hope to begin testing their system on patients in the near future. They also want to build chips that can draw energy from the patient's body, eliminating the need for a battery.

Other MIT researchers are working on implantable medical-monitoring devices, which would have to run with a battery that doesn't require recharging.

Associate professor Joel Dawson is working on a device that stores energy in an ultracapacitor, an electrochemical capacitor with high energy-density levels.

IBM Advances Ultrafast Memory Technique

IBM says that it has made important research advances into



IBM has developed advances in phase-change memory, an approach that proponents say will increase computer performance in ways that flash memory hasn't been capable of achieving. One of IBM's important advances, shown on this PCM chip, is a practical multilevel storage approach that increases memory density.

a potentially important ultrafast memory approach and that the technology could be ready for use in servers by 2016. Proponents contend that phase-change memory could increase computer performance in ways that flash memory hasn't been capable of achieving.

PCM stores the ones and zeros of binary data via the heating of chalcogenide glass, which changes the material's electrical properties. When chalcogenide cools quickly, its molecules stay in an amorphous state and the material doesn't conduct electricity well. However, when chalcogenide cools slowly, its molecules line up in a way that transmits electricity more effectively. Measuring the electrical resistance after the heating process determines whether stored bit is a one or zero.

Researchers have discussed PCM for about 40 years. Progress has been slower on using the technology in servers than in other devices. For example, PCM chips for cellular phones are already available, replacing flash memory. However, manufacturers see servers as perhaps the most lucrative PCM market.

Servers have been increasingly using flash memory, in the form of solid-state disks. However, these flash disks are costly and wear out faster than users would like. Enterprise-level flash starts to lose effectiveness after 30,000 write cycles. PCM can maintain performance to at least 10 million cycles.

IBM says it anticipates that rather than replace the faster dynamic RAM, PCM will work with DRAM, perhaps by caching data for reuse.

Company researchers say one of their big PCM advances is finding a practical multilevel storage approach that increases memory density.

With multilevel storage, a PCM system cools cells at various rates so that there are four different molecular states, which enables each cell to store two data bits rather than one as in the past. This doubles memory density and halves memory costs. IBM is working to increase density because flash memory can already store three bits of data in each cell.

IBM said it made the multilevel approach practical by avoiding drift, a problem that can increase error rates. With drift, a memory cell's electrical resistance changes over time, making consistent measurement difficult and potentially corrupting data.

IBM's approach measures the relative, rather than absolute, resistance of each cell, which, company researchers say, reduces errors to acceptable levels.

The company also wants to make PCM, which is potentially less expensive than flash, more cost-effective by building it with newer manufacturing processes that offer smaller feature sizes. Currently, IBM is using an older 90-nanometer process for PCM.

IBM says it doesn't plan to make PCM chips itself but instead will license its approach to other manufacturers.

Numerous chip makers—including Hynix, Micron, and Samsung—as well as various academics, are working on phase-change approaches.

Microsoft Technique Identifies Unused Wireless Frequencies

Microsoft researchers have developed a technique to determine whether users are transmitting over their licensed radio frequencies at any given time so that, when they're not doing so, unlicensed devices could utilize the spectrum.

Approaches such as Microsoft's SpecNet could become important as wireless usage grows rapidly. This is a particular problem because much of the wireless spectrum is owned by license holders, who control the resource whether they're utilizing it or not.

SpecNet would measure and then map where spectrum is and isn't being utilized. Unused frequencies could then be made available with the cooperation of the spectrum holders,

Microsoft's technique would employ a network of spectrum analyzers run by servers that issue commands via XML remote procedure calls over HTTP. The servers would be controlled by a master server that oversees data collection throughout the network.

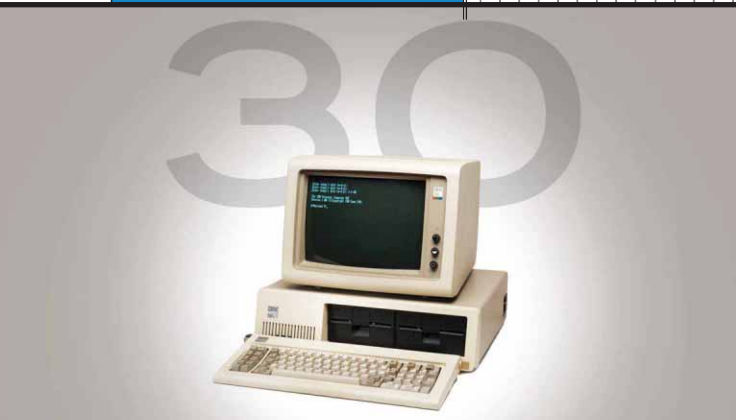
Researchers designed the system to conduct frequency scans rapidly and efficiently and to balance the workload among spectrum analyzers.

A major challenge for SpecNet is that spectrum owners would have to agree to participate in the project. Also problematic is the high price of the analyzers, which can cost up to $40,000 apiece. Microsoft researchers have said perhaps organizations with analyzers that aren't always in operation could donate their use for assigned time periods.

Another obstacle is that the US is the only country so far to approve the utilization of unused licensed wireless spectrum. C

Editor: Lee Garber, *Computer*; l.garber@computer.org

COVER FEATURE



A Personal History of the IBM PC

David Bradley

The development of the IBM PC was an exciting time—a brand-new design, a short design schedule, and lots of things to go wrong. An insider recalls the things that happened, the problems solved, and the interesting adventures along the way.

In 1974, Intel announced the 8080 microprocessor, the foundation for the era of personal computing. Widespread personal computing began with the publication of a cover story on the Micro Instrumentation Telemetry Systems (MITS) Altair in the January 1975 issue of *Popular Mechanics*. This kit included 256 bytes of memory and an 8080; MITS had expected to sell several hundred, but ended up selling thousands. It wasn't the perfect machine, but it appeared at the perfect time. The Altair initiated the Standard 100 pin (S100) bus, and launched clone companies building machines nearly identical to it. It also enticed two college students to write a Basic interpreter for the machine: Paul Allen and Bill Gates would turn that first program into a software empire.

In 1977, Apple announced the Apple II, perhaps the first appliance computer—usable by people not interested in building their computer from a kit. The introduction of the first spreadsheet program, a combination of VisiCalc and the Apple II, produced a system capable of real productive work, offering a capability that didn't exist on mainframes.

IBM AND PERSONAL COMPUTERS

IBM noticed personal computers as they began appearing inside accounts nominally powered by large System/360 mainframes. The first IBM response was the 5100 system, developed in Rochester, Minnesota. It presented a user interface based on one of two interpreted computer languages. The options were to purchase a system with Basic or APL or both.

This system was expensive (nearly $20,000 with 64 Kbytes of RAM in 1975) and anemic in performance, at least by later standards. However, it was an impressive step up in design. At its core was the IBM-designed Program All Logic in Memory (Palm) microprocessor, designed in Boca Raton, Florida. The Palm emulated a subset of the System/360 instruction set so it could run the System/360 APL interpreter. Compared to a real System/360, this emulation was exceedingly slow. But, amazingly, you could start a complicated APL program, come back a day later, and it would have run to completion with the same results as the System/360 execution. It was orders of magnitude slower, but it worked. The Basic interpreter was similar, using System/3 Basic.

Evangelists within IBM had followed the rise of microprocessors and convinced the General Systems Division that IBM needed to develop a product based on a vendor microprocessor. Doing so would be a significant departure from IBM's practice. Before then, every IBM computer had a processor designed by IBM. But vendor processors were now more powerful, comparably priced, and had support from a variety of tools and chips. It was a classic "make or buy" engineering decision, and, in this instance, the evangelists convinced IBM to buy.

Development of the resulting design, the System/23 Datamaster, began in early 1978 in Boca Raton, where I was one of the developers. We worked in offices carved out of a portion of a large leased warehouse. If the personal computer industry's norm was five guys in a garage, IBM's answer was 50 guys in Building 203. IBM chose the Intel 8085 for its stable architecture and the availability of support chips for the Datamaster. The 8085 was an extension of the 8080, notable for requiring only a 5-volt power supply. It was limited to 16-bit addressing and 8-bit arithmetic.

The Datamaster was to be similar to the IBM 5100, presenting a Basic interpreter as the primary user interface.

0018-9162/11/$26.00 © 2011 IEEE Published by the IEEE Computer Society

Simple file handling routines provided access to 8-inch diskette drives holding 128 Kbytes. The IBM-written Basic interpreter proved difficult since it had to conform to IBM's standards and run on the 8085. Consequently, development fell far behind schedule.

DATAMASTER LEGACY

Datamaster was very important for the IBM PC—even though it was announced just two weeks before the PC. Consequently, it was lost in the PC news. It's a tribute to IBM salesmen that they sold as many Datamasters as they did. But Datamaster was the starting point for many design points that ended up in the IBM PC.

When IBM embarked on the PC project, an obvious center of vendor microprocessor experience was the Boca Raton team that had developed the Datamaster. Bill Lowe, the system manager for Datamaster, was a strong advocate for the PC within IBM. And, perhaps because it was far from corporate headquarters in Armonk, New York, Boca was an excellent place to develop a system with little headquarters meddling and influence. So the IBM PC began in Boca, seeded with the engineers and programmers who had worked on the Datamaster.

The IBM PC began in Boca, seeded with the engineers and programmers who had worked on the Datamaster.

Perhaps the most important decision that Frank Carey, IBM's CEO, made was that the development would proceed outside the normal IBM process requiring adherence to corporate standards. We would still need to design, build, and test a quality product, but we would be answerable only for the final product.

Why did IBM choose the Intel 8088 for the IBM PC, setting in place half of the Wintel duopoly of the 1990s? The Datamaster experience with the 8085 was a strong component. The Datamaster team had become familiar with the Intel chipset and the available support chips. We had worked well with Intel. We had a huge number of Intel Microprocessor Development Systems (MDSs—otherwise known as Big Blue Boxes), and were familiar with using them. When we started the IBM PC with a one-year development schedule, choosing something familiar made sense.

The Datamaster had a couple of expansion slots that were essentially a demultiplexed 8085 system bus with some addressing extensions. These slots allowed for memory cards and some special I/O devices. In particular, we developed a synchronous data link control (SDLC) controller and a binary synchronous (Bisync) communications controller. These system interconnect methods have long since passed into the realm of the forgotten. But, when it came time to determine the system expansion bus for the IBM PC, we made it nearly identical to the S/23 bus. That would allow us to use the SDLC and Bisync adapters in the PC with minor modifications. So how many SDLC adapters for PCs do you think were ever sold?

The IBM PC also expanded the ASCII character set by filling the upper 128 positions available in a byte character. In keeping with the IBM PC's short development time, the added characters were selected during a single three-person five-hour meeting on a flight from Seattle to Atlanta. This group chose the characters necessary for the world's major languages, something well understood because of similar work on the Datamaster. Of course, others further reviewed this work, but very few changes resulted. The use of ASCII was somewhat controversial since IBM systems were exclusively extended binary-coded decimal-interchange code (EBCDIC), including the Datamaster. But we felt the need to move to a standard more familiar to the personal computer world.

THE IBM PC

I joined the PC development group in September 1980. Every morning for about a month, I met in the conference room off the main hallway with 11 other engineers who were working on a brand-new project, codenamed Acorn. I had heard a little about it, but knew none of the details. Now I'd be in the middle of it.

These 12 people were the beginning of the IBM PC hardware design team. There was a manager, a chief engineer, system board designers, video controller designers, a diskette controller designer, mechanical designers—"metal benders"—and printer and parts procurement. I was responsible for the basic input/output system (BIOS), otherwise known as engineering software. All of these engineers were from the Datamaster development group.

There is a myth that the IBM PC was developed by 12 engineers. Since success has many fathers while failure is an orphan, many people claim to have been one of the "original 12." I believe the myth arose when *Think* magazine (an internal IBM glossy-cover publication) ran an article celebrating the PC's first anniversary. The article mentioned 12 original developers without being specific about just who they were. And so, just about anybody who was around in the early days believes they were one of the original 12.

In addition to the group of 12 that I joined, there are other deserving claimants, which pretty much shows that there were lots more than 12 people who developed the IBM PC. First, in August 1980 there had been a task force—IBMese for a group with diverse skills that meets frequently and exhaustively on a specific subject to prepare recommendations for senior executives. This group had laid out the basics of the IBM PC and delivered a set of hardware overviews that provided the early design documents for the PC.

The task force had a varying complement, but the average probably fluctuated around a dozen. Also, when Don Estridge came onboard to manage the PC program, his inner circle was probably about a dozen. In addition to engineers, there was an early software group that began negotiating with software vendors (including Microsoft) about the content and quality of the programs they supplied that IBM would publish. There were more than 500 engineers, programmers, planners, marketers, managers, and, most importantly, manufacturers, by the time IBM announced the PC. So the myth of 12 developers is bogus, although with some fleeting foundation in fact.

And, at no time within the first year of the IBM PC, did we ever refer to ourselves as the "dirty dozen."

EARLY DESIGN

With our daily get-togethers, we launched into the new product's design. The meetings didn't last long—no more than 30 minutes. The goal was to keep everybody apprised of current activity and ensure that we weren't working at cross-purposes. The one-year development meant that early design had to move quickly and efficiently.

We had a preliminary design document from the task force, a kind of high-level block diagram. Much of the design was similar to the Datamaster. Our experience with the Intel chipset was exploited—same timer, interrupt controller, and parallel-port interfaces. We used the same diskette controller from NEC. We designed a different video adapter—two of them in fact—using the Motorola video controller. The 8088's additional address space made it desirable to include all video RAM within the processor address space, and we did. In so doing, we created one of the "holes" in the address space that would bedevil OS designers for the life of the PC.

In Datamaster development, we had done all programming on MDSs, storing everything on 8-inch 128-Kbyte diskettes. I was thrilled to discover that one of the first things that our lead engineer had ordered was a new MDS with a 5-Mbyte hard drive. The MDS unit itself was huge—about 2 feet square and 4 feet high. The hard file was actually a 14-inch removable disk cartridge. Five megabytes is small, but compared to the manipulations of diskettes, a hard drive was pure bliss.

Prototype development

To get the hardware design up and running, we built prototypes using wire wrap technology. A parallel activity was laying out the printed circuit board, but the finished card was still a month away. We needed a prototype to deliver to Microsoft for program development, and another to test the circuit design before committing the printed board.

If you've ever had a science fair project go bad, you can identify with the wire wrap prototypes. We built three of them. Two worked. Figure 1 shows the one delivered to Microsoft.

Prototype delivery

Along with two others, I delivered the first PC prototype to Microsoft. The contract with Microsoft required us to deliver a prototype "before December 1." Since December 1 fell on a Monday, we actually called and asked if they wanted us to show up on Sunday. They said Monday morning would be just fine.

On the Sunday after Thanksgiving, one of the busiest travel days of the year, we flew from Fort Lauderdale to Seattle with nine boxes of stuff—the disassembled prototype as well as some tools and test equipment in case we had to restore the system to functionality, something we expected with a fragile wire wrap board. On arrival, we rented a station wagon to carry everything (this was before minivans became popular).

Five megabytes is small, but compared to the manipulations of diskettes, a hard drive was pure bliss.

We showed up at the One National Bank building in downtown Bellevue at about 7:45 a.m. Monday with our boxes in tow. Steve Ballmer answered the door and showed us the small room (perhaps 8' × 12') where we were to set up our prototype. For security, this room was far from the normal development lab, across from the people who were writing the tech manuals for Microsoft products. We originally shared the room with bags of packing peanuts, since this was also their shipping area.

Thinking about that trip reminds me of a story indicative of IBM at that time. We took some test equipment with us, but didn't want to take more than needed. So we called the IBM Seattle branch office before we left and arranged to borrow an oscilloscope, just in case. We drove to Seattle Monday afternoon, entered the branch, found the customer engineering section, and said we were from Boca and wanted to borrow an oscilloscope. No problem; the customer engineer (CE) showed us the storage room that housed the equipment. As we headed for the door, I asked the guy, "Don't you want to see an IBM badge?" He said, "Okay." I showed it to him, and we headed back to Bellevue. They were very trusting back then. And we did return the oscilloscope.

During one of my later trips to Microsoft, I witnessed Bill Gates' personal involvement with the PC. In those pre-Internet days, we hand-carried diskettes between the two sites. The software group in Boca had asked me to pick

COVER FEATURE

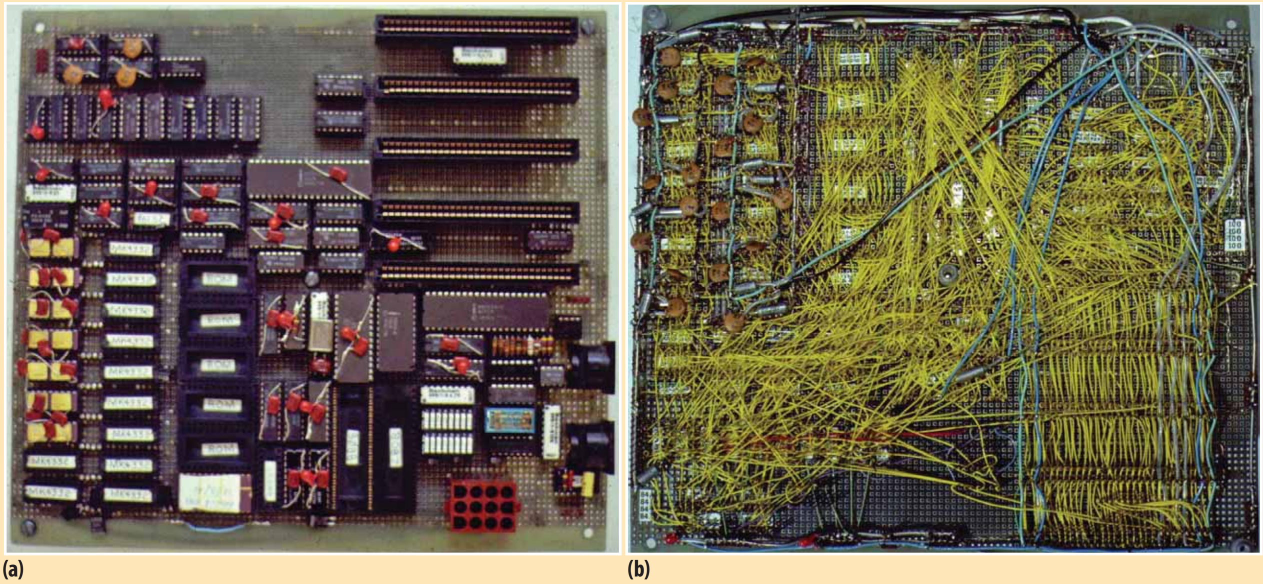


Figure 1. IBM PC motherboard prototype: (a) front and (b) back.

up the most recent version of the Basic interpreter. But it wasn't ready when I was done on Monday evening, so I arranged to stop by very early in the morning on my way to the airport. My Eastern Airlines flight left Seattle at 8:00 a.m. on its way to South Florida. Because the traffic along I-5 could be pretty brutal in the mornings, I was at the One National Bank building at 5:30 a.m.

I went to the PC room and found Bill on the floor, marking up a code listing. I asked for the diskette, but Bill said it wasn't ready; he had to make some changes that would take just a few minutes, or, in what would become traditional in the PC business, "real soon now." So I was looking at my watch, and Bill was making changes, rebuilding the interpreter, and trying out the changes. Finally, at about 6:30 a.m., I said, "I've got to go NOW." Diskette in hand, it took one hour to get to SeaTac. I just made the flight back home.

To highlight the graphics offered by Basic, Bill wrote the game called *Donkey* (DONKEY.BAS) that Microsoft distributed on a DOS diskette. In *Donkey*, you controlled a car driving up a road; a donkey would appear in one of the two lanes, and you pressed the space bar to make the car change lanes. The goal was to avoid the donkey and get your car to the end of the road. *Donkey* was the last complete program that Bill wrote for a product.

CTL-ALT-DEL

As things began to come together for the IBM PC, the design team had a development problem. Lots of new programs were arriving and being tested, and, as frequently happens, they would quit running, and nothing could revive them. When you're programming in assembly language, there are many opportunities to mess up. Unless you were working on a system board outside the case, the only way to restart the machine was to turn off the power, wait a few seconds for the capacitors to discharge, then power up the system again. You then had to wait for the power-on self-test (POST) to execute—which could take up to a minute if you had a lot of memory installed in the machine—then wait for DOS to boot (from the diskette, of course), and then you could restart the program under test and try to figure out what went wrong.

The simple solution would have been a reset button. But the PC's mechanical construction made it difficult to place one, and then there's always the problem of hitting it inadvertently. So we decided to use the keyboard.

I picked a set of keys that would be difficult to hit by accident. Since the BIOS had to fit in 8 Kbytes and was then closer to 9 Kbytes, it had to be a very short detection sequence. That meant that two of the keys would be shift keys, since the KB BIOS was already tracking them and recording the shift state in RAM. So I picked the two least-frequently used shift keys, Ctl and Alt, and Del—an action key as far removed from them as possible.

Figure 2 shows the PC's original 83-key keyboard (significantly different than today's keyboards). The single instances of Ctl and Alt are on the left side of the keyboard, and the single instance of Del is on the far right. Short of sitting on the keyboard, it's difficult to accidentally hit those three keys. I could have chosen "Enter" rather than Del, but Enter was a bigger key, more likely to be hit by accident, and Del is a better mnemonic for the drastic action that's about to occur.

Figure 3 shows the code to handle Ctl-Alt-Del, which was nine instructions and 30 bytes long. After detecting the two shift states and the Del key, it stored a flag in a special memory location, then jumped to the 8088's reset location. The POST code detected the special flag (0x1234 in the BIOS RAM area) after system initialization, but before performing the long memory test. I assumed that the memory had been tested when the machine was first powered on, and another memory test was probably unnecessary now. Plus it was a lot faster this way.

This wasn't the first time I had used a three-key sequence for a system function. I had written a small debugger for use in our Datamaster development (think of DEBUG on the original DOS diskette). A developer gained access to that debugger with a three-key sequence similar to Ctl-Alt-Del (although not identical because the keyboard was the same, but the interface and keycaps were not). So using a three-key sequence for reset/reboot on the IBM PC was a natural choice.

We originally intended Ctl-Alt-Del to be a development tool (like the Datamaster debugger) and not for customer use. But in the IBM PC's diskette-only user environment (if anyone reading this purchased a nondiskette IBM PC and used a cassette tape player for data storage, I'm sure you quickly repented of that sin), each application program came on its own diskette. The "install" process usually required transferring the DOS boot image to the application diskette. To run that application, you booted from the diskette. The developers who were testing the software and writing the publications quickly learned that the easiest way to reboot the PC with a new application was Ctl-Alt-Del, and it was from them that the command "escaped" into general use.

Ctl-Alt-Del was a simple solution to a problem. It wasn't complete, since if the failing program managed to disable interrupts (single-byte instructions on the x86); nothing that happened on the keyboard went to the BIOS. But Ctl-Alt-Del worked in most cases. And since it was just one of hundreds of little (and big) problems that we had to solve before IBM could announce the PC, after 10 minutes of design, coding, and testing, it was time to move on to the next problem.

Figure 2. IBM PC original keyboard. The single instances of Ctl and Alt are on the left side of the keyboard, and the single instance of Del is on the far right, providing a set of keys that would be difficult to hit by accident.

FCC CERTIFICATION

The IBM PC was going to be used in the home, unusual for any IBM computer at the time. The Federal Communications Commission had strict standards for electromagnetic emissions from any device for home use, and the PC would have to conform. This was a technical challenge, made even more difficult because the Boca Raton laboratory didn't have a chamber suitable for electromagnetic compatibility (EMC) tests. Fortunately, open field testing was an alternative. However, if you start in Boca Raton and head away from built-up areas, you end up in the fringes of the Everglades. The EMC engineers would come in from a day of testing sunburned and smelling of insect repellent. But they didn't have to wear the shirt and tie that was the norm for IBM engineers at that time—they wore T-shirts and shorts.

```
EA6C F606170008      R         TEST   KB_FLAG,ALT_SHIFT   ; ARE WE IN ALTERNATE SHIFT
EA71 7503                      JNZ    K29                 ; JUMP IF ALTERNATE SHIFT
EA73 E98F00                    JMP    K38                 ; JUMP IF NOT ALTERNATE

                               TEST FOR RESET KEY SEQUENCE [CTL ALT DEL]

EA76                     K29:                             ; TEST-RESET
EA76 F606170004      R         TEST   KB_FLAG,CTL_SHIFT   ; ARE WE IN CONTROL SHIFT ALSO
EA7B 7431                      JZ     K31                 ; NO_RESET
EA7D 3C53                      CMP    AL,DEL_KEY          ; SHIFT STATE IS THERE, TEST KEY
EA7F 752D                      JNE    K31                 ; NO_RESET

                               CTL-ALT-DEL HAS BEEN FOUND, DO I/O CLEANUP

EA81 C70672003412    R         MOV    RESET_FLAG, 1234H   ; SET FLAG FOR RESET FUNCTION
EA87 E9D1F5                    JMP    START               ; JUMP TO POWER ON DIAGNOSTICS
```

Figure 3. **Assembly code that activates Ctl-Alt-Del keystrokes.**

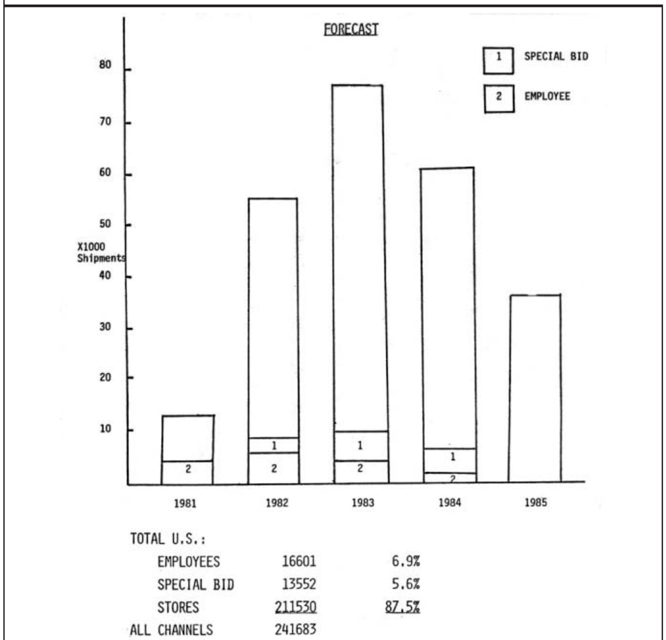


Figure 4. Preliminary sales estimate for the IBM PC, April 1981. The forecast was developed to convince executives to fund a major purchase of parts so that PC volume production could begin in the third quarter of 1981.

IBM PC FORECAST

Every business attempts to forecast its sales so it can plan for capacity and pricing. The IBM PC planners had to do this so they could persuade the IBM Board of Directors to fund a major purchase of parts so that PC volume production could begin in the third quarter of 1981. So they prepared an estimate and summarized it to the board in April 1981 in the chart that Figure 4 shows. (Note that this chart was hand-drawn with text added using a Selectric typewriter. There was no PowerPoint in 1981.)

The estimate was that in the five-year lifetime of the IBM PC, sales from all sources would equal 241,683 units. This forecast is noteworthy for two reasons. The first is its precision: six significant digits. Knowledge like that should make it easy to plan parts purchases. The second is its incorrectness: over the PC's five-year lifetime, IBM sold approximately 3 million systems, 250,000 in one month alone in 1984.

However, when IBM PC planners took this forecast to the corporate staff, they were admonished for suggesting such large sales volumes. Corporate staff suggested significantly reducing the number.

INITIAL SOFTWARE

When IBM announced the PC on 12 August 1981, it also announced software, both operating systems and applications. Everyone remembers PC-DOS, the diskette operating system for the PC developed by Microsoft. But there were two other choices: CP/M-86, the upgrade to the original CP/M for the 8086 architecture; and the University of California, San Diego's, p-System, an interpretive Pascal system.

Of the three OSs, only PC-DOS (renamed MS-DOS when it appeared on other manufacturers' machines) succeeded, eventually becoming Microsoft Windows, the other half of the ubiquitous Wintel duopoly of the 1990s. The other two disappeared. Simple economics determined the winner—MS-DOS sold for about $40, while CP/M-86 and p-System were about $400. After spending $2,000 to $4,000 or more for the IBM PC hardware, customers chose the cost-effective OS and made it the industry standard.

By the way, if you open the DOS window and use the command line input, the simple commands like Dir and Copy go all the way back to the original CP/M offered for machines like the MITS Altair.

IBM published seven application programs. This small but comprehensive set got the software business started:

- EasyWriter, a word processing program that included the nifty symbol of a musical note to indicate the end of a line (actually the ASCII CR character);
- VisiCalc, the original spreadsheet program;
- three business-related programs—Accounts Payable, Accounts Receivable, and General Ledger—by Peachtree Software (after all, the name of the company is International Business Machines);
- Dow-Jones, a communications program for tracking a stock portfolio; and
- *Adventure*, a text game played in Colossal Cave, including a maze of twisty little passages, all alike.

This offering had something for everyone: business apps, the personal productivity offerings that were the raison d'être for personal computers, and a game. Although VisiCalc was well-known, EasyWriter was one of many word processing programs. The author of EasyWriter was John Draper, also known as Cap'n Crunch for his discovery that the toy whistle offered as a prize in the cereal boxes would generate the 2,600-Hz tone necessary to gain free, illicit access to the long-distance telephone network. Cap'n Crunch was one of the premier phone phreakers of the 1970s, and the author of an IBM-published application.

I was a beta tester for *Adventure*. I still have a map of Colossal Cave, drawn on a 24" × 30" sheet of poster paper, in my notes.

Since the IBM PC would be serviced by IBM CEs in some instances, the development team had to provide hardware diagnostics for their use. That was the reason for the intricate series of beeps and characters written to the display during the POST. Each beep or character signified another successful test so the diagnostician could follow along and decide what had broken. Every customer got a *Guide to Operations* that described the system's basic operations and diagnostics. Customers also could purchase the *Hardware Maintenance and Service* (HMS) book that contained more extensive diagnostic facilities, including a special program to test all of the system and the adapters. And since it was possible to purchase a PC without diskette drives, there was a cassette tape inside the HMS book for those systems.

IBM PC OPENNESS

The key to the IBM PC's success was its openness. Even powerful IBM could not have assembled the resources necessary to develop all of the hardware, and especially all of the software, that made the IBM PC such a successful product. By getting the industry to participate with the PC, IBM assured its success. The Apple II was an open system. The technical manual for it had the firmware listing, system schematics, processor description, and lots of other technical detail (www.scribd.com/doc/24964/Apple-Reference-Manual-January-1978). In 1980, that was the standard for an open system.

So IBM chose to make the IBM PC an open system. With an Intel microprocessor and a Microsoft OS, the basic components were in place to allow others to participate in the IBM PC business. The *IBM Technical Reference Manual* (TRM) contained complete schematics for the system and adapters, assembly code listings for the ROM BIOS, and descriptions of all the I/O functions on the system board. Apple's approach with the Macintosh, which was not an open system and was very tightly controlled, stands in contrast to the IBM PC approach. Apple's users faced a less-confusing system in which everything worked in much the same manner, while the IBM PC had many more hardware and software offerings.

The TRM is also interesting for what it didn't contain. IBM didn't publish any real architectural description of the developer interface. Without those guidelines in place, creative people used just about every possible bit and byte in the PC. As an example, even though the entry points of the BIOS routines are accessed through a vector table, there was a program that used a routine's actual physical address. We discovered this when we found that the IBM PC XT prototype wouldn't run that program. We had to go back into the XT BIOS and rearrange the code so that it maintained the entry-point addresses.

A notable success story arising from the IBM PC's openness was Tecmar, a small adapter card company in Cleveland, Ohio. Within six weeks after delivery of the first IBM PCs, Tecmar announced 20 different adapter cards. True, the designs were adaptations of their current offerings to the IBM PC's bus, but it demonstrated an extent of offerings that was far beyond what IBM could afford to do.

Many companies, both hardware and software, jumped on the IBM PC bandwagon. By the PC's first anniversary in August 1982, a trade show devoted to it filled a large ballroom with vendors offering compatible hardware and software. *PC Magazine* came into existence around this time, eventually becoming a bimonthly publication the size of the Manhattan phone directory.

IBM SALESMANSHIP

In November 1982, I made a sales call to the Rose-Hulman Institute of Technology in Terre Haute, Indiana, along with IBM salesman Larry Byrnes. Our goal was to convince the engineering faculty to purchase and use IBM PCs in its labs and curricula—pretty standard today, but a revolutionary step at the time. My task was to describe the system architecture, and Larry was to demonstrate the bells and whistles.

The IBM PC had a very modest sound-generating capability compared to today's systems, but the design team was very proud of it. So much so that there was a demo program on the DOS diskette to show off the sound capabilities. Users could select a song (all of which were in the public domain, thanks to the crack IBM legal staff), then a musical note would dance around a keyboard while the song played through the 2-1/4" speaker included in the system.

Larry wanted to demo the sound system, but for a group of 30, the speaker just wasn't loud enough. So he had connected the cassette tape player output (which used the same tone-generation circuitry as the sound system) to a small guitar amplifier, which he placed beneath and behind the table the PC was sitting on. At the conclusion of his presentation, Larry brought up the sound demo program and invited a faculty member to select a tune. But it didn't matter which button was pressed, only one song would play, Sousa's "Stars and Stripes Forever."

Within the amplifier, Larry had installed an electric car radio antenna, which he could activate using the PC system board relay that controlled the cassette tape motor. As the rousing Sousa march began filling the room, the radio antenna began rising behind the IBM PC with a small American flag attached to it. This brought applause from the viewers, and we made the sale.

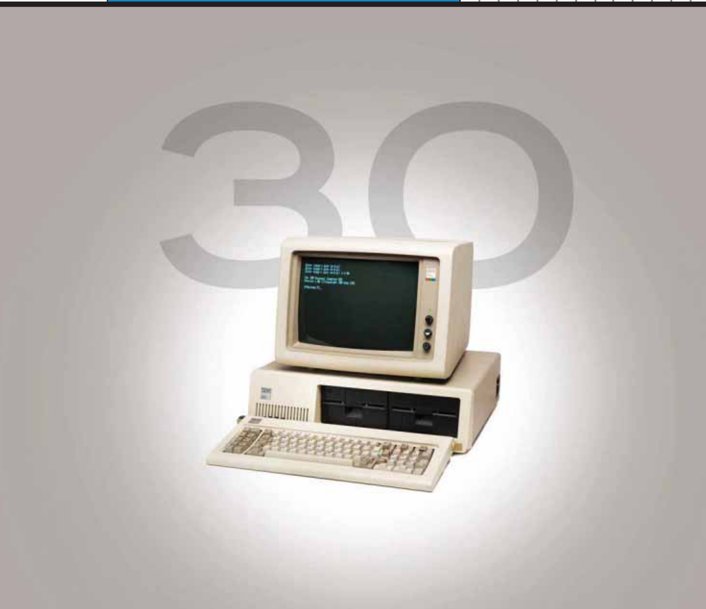
As much as the development team would like to think of the design as being the most important ingredient for the IBM PC's success, you can't forget the salesmen.

Despite its bland beige covers, the IBM PC was an outstanding success, measured both in sales and impact on all future personal-computer offerings. The IBM name meant stability and respectability in a market perceived by many as a hobby. Its design was new but not revolutionary, with good performance, and it instantly became an industry standard. Openness encouraged broad participation, with competitive hardware and software offerings across the industry. Most importantly, the PC was from IBM. In the 1980s, you would "never get fired for buying IBM." ■

***David Bradley** retired in 2004 after more than 28 years with IBM. He has also been an adjunct professor at Florida Atlantic University and North Carolina State University. Bradley received a PhD in electrical engineering from Purdue University. Contact him at bradley.dave@nc.rr.com.*

Computer | Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page | Qmags THE WORLD'S NEWSSTAND®

COVER FEATURE



IBM PC Retrospective: There Was Enough Right to Make It Work

Greg Goth

Mark Dean, a participant in the original IBM PC design team, looks back 30 years to review the events that led to the design of a very "un-IBM" computer—a small device that used hardware and software from outside vendors and made the logic and BIOS code available to anybody.

Computer: In the late 1970s, you could walk down the main aisle in the Poughkeepsie IBM plant and see mainframe after mainframe being constructed; there was a sign hanging from the ceiling over each one with its customer's name. All those customers were large institutions—the Department of Defense, Department of Labor, banks, and so on. In 1980, your team was charged with creating the antithesis of that computing model. Do you remember how the project began?

Dean: We proposed that we could build something that would move computing from the mainframe model to a personal organization and productivity model. We had dabbled in this space with machines such as Displaywriter and Datamaster, so we kind of knew how to do this. But we hadn't done it so that my mother could buy a machine and use it. It wasn't personal yet. It was focused on something like word processing, which Displaywriter was, or some type of calculating capability, which was what Datamaster did, and providing an interface to the mainframe. We wanted to operate locally. We said, "We can build this. There will be a set of applications that will be key. So you should let us do this."

And management said, "Okay, fine. It won't hurt anything. It's not going to mess up our business. You're not trying to replace us. You have 10 or 12 people. When you find it's really not much of anything, you'll come back and we'll do real computing." That was the attitude. They weren't afraid. They thought it would be, maybe, interesting. It might actually be a follow-on to Displaywriter and Datamaster. We'd sell maybe 100,000 or 200,000 units, and then we'd keep doing computing as we had always done it. So that's why they left us alone. We were off the radar.

Computer: How familiar were the team members with the guts of other personal computing platforms?

Dean: Oh, we were true nerds. I was doing my master's thesis on what I called a high-performance graphic terminal, basically using the same components being used to build various versions of PCs. Companies essentially had built toys to that date. What hurt them was that everybody had built computing similar to the way it was done with mainframes—they built the hardware, and it was up to the buyer to program it. They didn't deliver an application or an operating system. They delivered just enough to turn the machine on and have the lights blink. Nobody had combined a set of applications with the hardware so that when you bought it, you could just load it and start to do work within a day. That was a big deal, and there was a pent-up demand for it.

Computer: Did you consider a particular machine built by somebody else, like the Apple II, as a central competitor?

Published by the IEEE Computer Society

Computer Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page

Dean: We were aware of what was being built. We knew how successful or not those things were. So we knew what not to do, and we kind of knew what was working. Apple was probably the dominant player because they were offering a little more than just hardware, but they weren't focused on personal productivity as I remember it. They didn't have a suite of applications that drew people in. In addition, Apple was proprietary. Nobody else could build anything that would either plug into it or could duplicate it. That was the main reason the IBM PC won over the Apple—other people could build the exact same thing. We provided enough information for people to write applications for it, and they could build hardware to add onto it as well.

***Computer*:** Were you ever aware of some general IBM dissatisfaction with the "Boca cowboys" for an approach that countered the traditional IBM development cycle strategy? Or again, did they just let you go?

Dean: They mostly let us go. I remember being questioned about our go-to-market strategy, the volumes, and the actual ability to build a quality machine. People would say, "The way you guarantee quality is different than the way we've been going about it for the mainframe. It's going to break all the time and not be a very reliable system." We said we didn't agree with that.

***Computer*:** How were you approached and why you?

Dean: Like most of the 12 or 13 people on the design team, I just happened to be in the right place at the right time. I had done some graphics and display cards for Datamaster, and they needed that kind of design work. I developed the color graphics card and the monochrome card. I became the video and graphic designer. I was in heaven. It was something I loved doing. I did two or three different designs at the same time, and this was one of them.

***Computer*:** Did you ever meet any of the Microsoft guys?

Dean: Other IBMers like Lew Eggebrecht were closer to Gates and the Intel team. I was in the next level down, so I would interact with the guys below that. However, there weren't too many people below Gates, because there were only two or three other people in Microsoft at that time. Intel also had a very shallow organization in terms of layers at the time. So it was a pretty tight-knit family. In general, the IBMers worked pretty closely with the Intel and Microsoft guys.

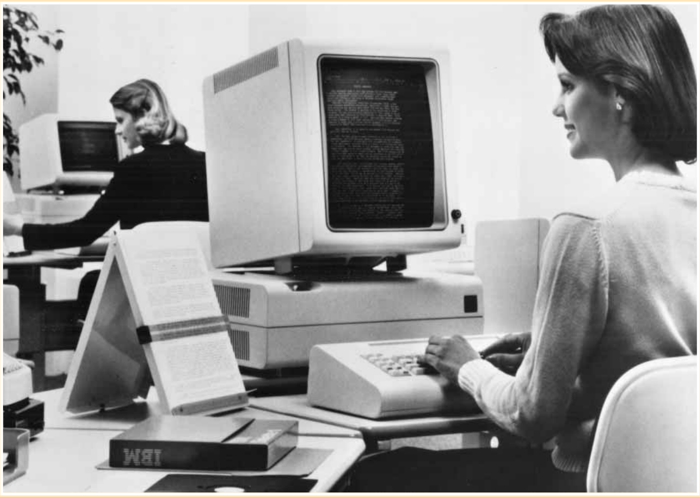


Figure 1. Displaywriter, introduced in June 1980. Innovations introduced in the desktop text-processing system included the ability to store a document, recall it for review or revision, and check the spelling of approximately 50,000 commonly used words, novel features at a time when most documents were created, formatted, and revised on manual or electric typewriters.

Figure 2. Datamaster, introduced in July 1981. Announced one month before the IBM PC, the System/23 Datamaster offered a viewing screen, keyboard, and diskette drives packaged in a single desktop console to support word processing and data processing in a machine that gave small businesses the big benefits of information processing.

DESIGN CHOICES

***Computer*:** IBM had its own microprocessors then as now. So why the choice of Intel over the IBM chip? Was there talk about using the homegrown 801?

Dean: At the time, the notion of a CMOS-based "microprocessor" was a fairly new idea. Remember, the mainframes at the time were still bipolar. We had just started building processors that were 801-ish, RISC-based. That was kind of a new concept as well, not completely figured out. Basically, IBM didn't have a processor that could fit into a box smaller than a refrigerator. But we wanted to build something smaller than a refrigerator, and we knew we needed to use something we could power by plugging it into a wall socket, not a 220-volt outlet. So there were constraints.

Datamaster and Displaywriter were built using off-the-shelf Intel processors. Although we used Intel processors, we actually evaluated and did prototypes with Motorola processors as well. In fact, a group of us thought the Motorola processors might be a better way to go. I had built a prototype of the 6809, which was one of the competitors of the 8088. But going with Intel was a matter of choice, probably because that's what we were comfortable with. The graphics card, which I did, was Motorola-based. So we were the first to blend Motorola-based chips with Intel-based chips in the same system.

> **We wanted to build something smaller than a refrigerator, and we knew we needed to use something we could power by plugging it into a wall socket, not a 220-volt outlet.**

Computer: Was it hard to do that?

Dean: Actually, it was pretty straightforward. We had gotten feedback from both Intel and Motorola that it couldn't be done, but we wanted to use the best of what each had to offer. We thought the Motorola graphics chips worked well. We thought the Intel processors would work well enough for us because we had used them, and we thought the 8-bit 8088 was all we needed in the first line. In fact, back then, that was a big deal. If you could get 64 Kbytes of memory, that was all you would ever need. By combining it with a processor running at 1 megahertz, we thought, "How is anybody ever going to use all this processing power?" We had to make those kinds of choices if we were going to build something in 12 or 13 months.

IBM had much stronger technology, but we didn't have anything available off the shelf. The three choices were Motorola, Intel, and the Zilog 6502. We built prototypes from each of those, and we closed on the one we felt would get us there.

Programming tricks

Computer: Is there anything in particular that you remember regarding the programming tricks between the Motorola and Intel chips?

Dean: Obviously, each one had a different clocking scheme. We had to adapt the clocking schemes so they would sync up and communicate. To make it work, we had to adapt the processor requesting signaling to the I/O signaling that was on the graphics controller and use some synchronization best practices. Once we knew the clocking and signaling from the processor and knew how the I/O devices wanted to respond, we needed a set of interfaces in between that would do the translation.

It was actually more than just Intel to Motorola chips—we also had the floppy interface, which was done by a different vendor, and the interfaces to the printer and cassette. We sold the original PC without a floppy, and users could buy a cassette deck. So we had to build that interface out of something else.

All these interfaces were not necessarily designed to work together. The ISA bus became the point of translation. That was the standard that everybody had to work off. Build to this, and it will work.

Inclusive design

Computer: Do you think it worked so well because you brought so much in from outside?

Dean: Whether we did it on purpose or not, the bottom line is that we were successful because the PC was inclusive. It enabled other people to participate in the industry. We essentially sold a platform that offered a base where everyone else could jump in. No one else had built something that allowed everybody else to play. We gave away logics and the BIOS software, which was written down in the technical reference manual. Everything was in that book. You could buy the exact same parts, program them the exact same way, and build the exact same thing, and that's what a lot of people did. That not only allowed people to build IBM-compatible machines, but also gave them enough detail to build adapters and other devices that would plug into the IBM machine.

Computer: Deep in their heart, everyone believes they could build a better microprocessor than Intel and a better operating system than Microsoft. Were you happy to outsource those big risky items, or was the outsourcing the product of that short timeline in a company that wasn't particularly good at guerrilla product development?

Dean: We always felt we could do it better. We felt we could build a better processor, and we tried, long term. We always felt we could build a better operating system, and we tried. Engineers always want to build the ultimate thing. It's just our nature. What we forget is that my mother doesn't care. All she cares about is "Just give me what I need to get something done." And so that's what we built.

If you look across history, it's almost never the case that the best technology wins. It's the technology that addresses the problem in a timely fashion that wins, and that has happened over and over again. There were better machines on the market, I have to admit. In the first 10 years of the PC, companies were building other things, not compatible with the PC, that you could argue were better. But they didn't make a difference—they weren't better enough.

Design tradeoffs

***Computer*:** Were there any interesting technical problems designing the PC? Difficult tradeoffs? Anything that stumped you?

Dean: We had had so much experience building systems like this, we kind of knew how to do it. I have to admit that some of the things we did were guesses. We didn't quite know how people were going to use the machine. We could only imagine what they might want to add. Did we define what became the ISA bus correctly enough so that they could add a new feature?

There were some interesting firsts. For example, we were among the first to build a color graphics interface. We could have done a lot of different ones. In fact, in the first designs we had twice the resolution we needed. And we said, "Well, that's too much to program. Do we really need that much?" So we cut it back because it required twice as much memory, and we thought the cost was too high and the card was too big. We worried about cooling and power supplies. We worried about whether we could support this new 5-1/4-inch floppy format. To date, the smallest floppy we had done was an 8-inch.

***Computer*:** The lore is that there are nine fundamental patents in the PC, and you hold three of those. What are they and what do they cover?

Dean: Two of them had to do with the way you put pixels on the screen—the graphic interface and color and pixel creation and control on the monitor. The third had to do with refreshing the memory interface. While we patented nine things that we thought needed to be patented, we didn't patent three times that—additional design points that probably had as much value as those patents. Those nine patents, including my three, were important, but I'm as proud or maybe more proud of other design points I felt were as key to the success. Those nine patents provided a definition of what became known as IBM PC-compatible.

Figure 3. IBM Selectric typewriter. The IBM PC keyboard differentiated it from all the other clones because it was built on the requirements set for the Selectric. Introduced in 1961, the IBM Selectric represented a technological breakthrough that revolutionized the typewriter industry.

***Computer*:** Did the ISA bus accrete around the microprocessor bus or was it designed to allow other companies to play? Also, was the fact that it was not plug-and-play due in part to market immaturity?

Dean: Plug-and-play wasn't something we had in mind. We had talked about PnP, and we knew that at some point we would be able to do it, but we didn't have enough technology capability or the need to do PnP at the time. It would have taken too much hardware to support PnP, which was a fairly new concept at the time we built the first ISA-based machine.

The machine was actually designed to allow connecting different devices. It wasn't designed just off the processor interface. In fact, as time went on, it didn't resemble the processor interfaces very well at all. Processors were designed mostly to connect to memory. We had to mostly connect to I/O. Although it was possible to put memory on the ISA bus, it was mostly designed for I/O. It was biased to allow scaling the machine by adding other types of devices. We included features that only a few people really took advantage of.

***Computer*:** What was the source of the PC's keyboard?

Dean: The Selectric keyboard was the industry standard for typewriters. People loved that keyboard. They thought it was the greatest keyboard ever invented. The PC keyboard was a big hit because it was built based on the requirements set for the Selectric. In fact, the keyboard differentiated the IBM PC from all the clones because nobody

COVER FEATURE

could build one of similar quality. It had the same feel, that little "click," which we included in the design because people liked it so much. And it had to have a certain travel, which meant the keys had to be a certain length. People didn't like the chiclet keyboard on the PCjr because they had gotten used to the great keyboard on the PC.

Computer: What artifacts do you see from the original PC when you look at a modern PC? Is there anything that stands out?

Dean: We still haven't gotten past hitting "Start" to turn the machine off. There's still Ctl-Alt-Del, and it still brings up the task manager. It used to just shut things off, but now it takes you to a control point. But it's still a key sequence. If you get a screen error, that DOS information still flashes up in front of you. In fact, some power-on key sequences still take you to that background process. So there are still some remnants. In fact, some retail stores still have DOS screens on their cash registers.

We have yet to build an input capability that is more efficient than the qwerty keyboard.

But in general, the only thing that's causing us to hang onto these laptops and the old model is the keyboard. It's amazing that it has survived so long. The design was actually downfunctioned when it was invented. They scrambled the keys to slow the user down because the mechanical device couldn't keep up, but it has survived. It's another one of those unoptimized design points, but it won. We have yet to build an input capability that is more efficient than the qwerty keyboard.

MARKETING STRATEGY

Computer: What aspects of the PC system design are you most proud of? What would you like to go back and redo if you could?

Dean: Despite all the things that we screwed up, I would build everything exactly the same. Although there were lots of mistakes, there was enough right to make it work. I'm afraid that if we did it differently, we might miss on some of those things that we now call right. They probably enabled another breakthrough, something we weren't expecting. In some cases, if we had built it a different way, maybe more optimized—the bus, the graphics interface, or something else—people might have opted not to do enhancements. There's very little, if anything, that I would suggest we do differently.

Now, if you asked me if we should have done our product plan differently, I'd have to admit that we lost sight of why the PC had become successful when we went to the PS/2. To enable continued growth, we should have continued with the model of building it so that other people could play. That would have allowed us to stay in control of the market. When we did the PS/2, we lost control.

Computer: Whose idea was the change in strategy?

Dean: There was a group of us who wanted to build a follow-on to the PC-AT that was directly compatible rather than something unique. In fact, the proposal was that we build both: we could move the higher-end machine to the next generation, but the lower end would continue on a path compatible with what we had built. For example, keep the ISA bus, just change it in increments. But there were others who said, "No, we're going to shift everything over to the microchannel." We shouldn't have done that. It wasn't the right approach.

Computer: During the early and mid-1990s, there was a lot of contention between OS/2 and Windows, which the IBM technology also lost in the marketplace.

Dean: The difference was, we needed to be clear about the market we were targeting so that we could avoid overdesigning for it. To be honest, the OS/2 was a better operating environment. It was reliable, bulletproof. But my mother didn't need that bulletproof operating system. She didn't want to wait 15 minutes for it to boot.

Nevertheless, we insisted on going after that personal market with an operating system that was really built as a bulletproof enterprise-scale product. OS/2 was great, but it couldn't compete in the market we tried to push it into. We were trying to compete in the same market as the Apple product, but the OS/2 had a tough time competing with that kind of user interface. Windows could kind of deal with it, but it still wasn't quite as good. To be honest, Apple was much better at it.

PRINTER TECHNOLOGY

Computer: When you were designing the original PCs, did you envision any particular workloads and application environments? The early promotion of the PC was for accountants and scientists, not writers. Did anybody see the eventual primacy of word processing, or was that just another application?

Dean: I think we did see the primacy of word processing. It didn't seem as compelling as the spreadsheet because of the quality of the printing. The best product available at the time was a dot-matrix printer. I could write a letter, but I didn't have a printing mechanism that made it look like I had typed it.

Computer: Was there any talk internally about pushing print technology faster?

Dean: Sure. IBM was in that business. We built laser, dot-matrix, and ink jet printers. In fact, my first design project for the PC was a controller for the dot matrix printer. I remember driving that print head so hard it buried the needles into the roller; I had to pry them out. Although I didn't really know what I was doing, I learned a lot from that experience. But IBM was in that business. It built printers for mainframes that could shoot the paper out so fast that it would stick it against the ceiling until the machine stopped printing.

Computer: Why did IBM get out of that business?

Dean: The margins, the returns. Other companies started to dominate the business. During its history, IBM has tended to focus on highly differentiated, high-margin segments of the business, and that has proven to be good for us.

'PC-FUL' COEXISTENCE WITH MAINFRAMES

Computer: Did you use the PC in your own work, or did you remain on workstations?

Dean: The very first prototypes that came out of the labs were ours to use. We designed it somewhat for us, so of course we wanted to have the first ones. If it didn't work for us, it wasn't good enough.

Computer: What was the reaction in the rest of IBM to the PC's runaway success? Did the mainframe guys acknowledge you had changed the way business was done? Were they resentful, or did they do their thing and you did yours?

Dean: They did their thing, and we did ours. The mainframe remained differentiated as our high-margin business, and it had a place that nothing that we built or anybody else has built could address. It was unique. The mainframe might not be as ubiquitous, and maybe not as many people have touched it. But without it, we would be nowhere near where we are today. For example, financial institutions wouldn't have the machines needed for banking processes.

The PC changed a lot of peoples' lives. It enabled computing to be done closer to an individual. So it was important, and I would say it had as much impact as the mainframe. But they were two different things supporting two very different environments, and we needed both.

Computer: The IBM PC was generally known as being very reliable, and IBM dominated the PC industry for almost a decade after its introduction. What was it that led IBM to de-emphasize and then sell the PC line to Lenovo?

Dean: IBM has always been in higher-margin businesses. It has never focused on being successful with consumer products. IBM still derives revenue from the PC through software and services. The company also provides support for emerging devices. You might say IBM doesn't make any money off of cell phones or tablets. But it will make a lot of money providing the back-end infrastructure and services for those devices. While it will never abandon the concept of supporting devices, building them is not high on its list.

The mainframe remained differentiated as our high-margin business, and it had a place that nothing that we built or anybody else has built could address.

Computer: Do you see any irony in the fact that when you designed the PC, you wanted people to play, but when they did, margins went down?

Dean: No. It was the right thing to do. We made some mistakes from a business standpoint over time, thus we lost market share, while other companies had room to grow. We might've wished we could've stayed more competitive. But even if we had done that, the market was going to go where it is—low margin, high volume. That's what the world needed.

Computer: Had IBM not done the PC, had the Apple model prevailed, would the industry have turned out differently?

Dean: It wouldn't have been nearly as pervasive. It's clear that proprietary designs don't foster significant industry growth. I've learned that if I want to drive more business, I need people playing in the industry. I need to compete. Much more revenue returned to IBM because everybody else was playing. We just needed to keep competing. The entire industry got much bigger than if we had tried to do it by ourselves. Apple never would have come close to creating an industry even one-tenth the size. Apple has a certain size business with the iPad and iPhones, but it can only do a certain size until another company does something on top, like Google with Android. It'll be interesting to see what will happen with the Android devices. Will they drive more volume than Apple devices? Probably. We'll see.

CHANGING THE WORLD

Computer: What is the "IBM PC technology" of today that's significantly underappreciated but which will change the world?

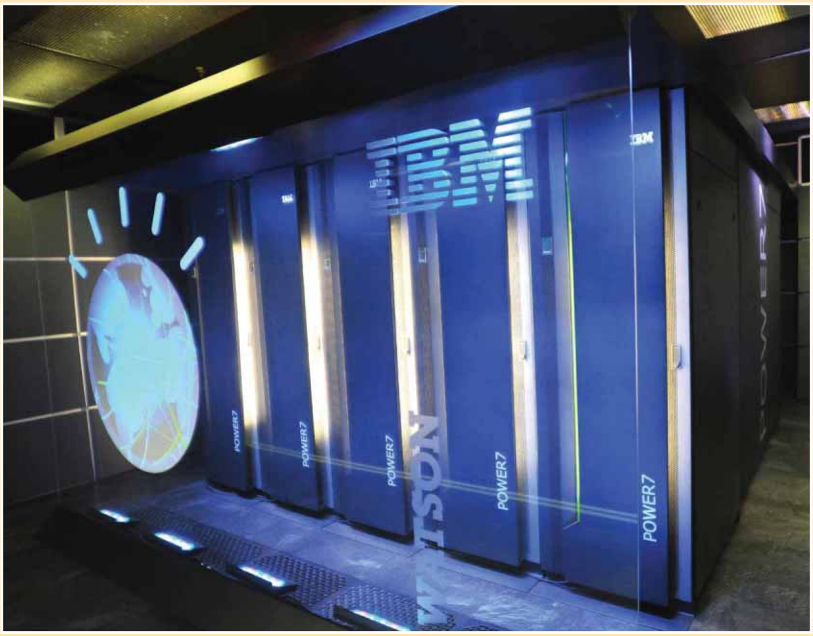


Figure 4. Watson question-answering computing system. Powered by IBM POWER7 processors and its DeepQA software, Watson is a workload-optimized system that can answer questions posed in natural language over a nearly unlimited range of knowledge.

Dean: There are little-known breakthroughs that will change the world, such as cognitive computing, a new computing paradigm that doesn't use the von Neumann model. The technology that we showed with Watson on *Jeopardy* will be more ubiquitous than PCs because it will be integrated into our daily lives. It may not get the play PCs did because it won't be as visceral, we won't feel it as much, but it will be bigger than search. If search was a change-the-world kind of thing, Watson's going to be bigger.

Computer: What do people never ask you about that you find important about the PC and its history? What myths haven't been corrected in 30 years?

Dean: There's a lack of knowledge about why the PC was successful. It wasn't the design points or the great technology. It wasn't that we had visions of changing the world. It was the fact that we got lucky enough to build the right thing at the right time, with the right set of functions, and accidentally allowed everyone else to play. We didn't put the logics and software in the technical reference to enable this great industry. We did it to allow the technician to repair the PC. Fortunately for us, it enabled others to build the same thing—"I can do this; this isn't rocket science. In fact, they even gave me the BIOS." In those days, there were so few lines of BIOS code that it actually was possible to list them on a piece of paper.

Computer: Looking back 30 years, have we come full circle? Apple is still known for sleek, intuitive end-user devices. IBM, which saturated the world with the PC, is now back in the news with Watson, a machine of immense capabilities that is clearly not meant to sit on someone's desk.

How do you see the synthesis of these more capable end-user devices coupled with the power behind Blue Gene and Watson as building on the legacy of what you did in Boca?

Dean: Those of us on the original design team who are still around are getting close to retirement, but we bring some institutional memory to the process. We still remember what worked and what didn't. We're actually building off all that learning of 30 years ago about what is and what isn't important. We've matured. We've gotten over the notion of "It's mine, I have to do it all." The notion is, "Can I deliver capability to a person or company that will help them differentiate themselves or their business?" Now we realize that we don't have to give them the device. We don't have to own the network. We can deliver capability without doing that.

I think we're seeing things repeat. There's a proliferation of PCs and handheld devices in the developed world. We haven't yet seen a proliferation of these devices in the developing world because they're starting at a different point. Nevertheless, their needs and wants are the same as those in the developed world a few years ago. We'll have to do it slightly differently, but we have to repeat our success. I hope we'll leverage what we learned 30, 40, and 50 years ago and ask how we can satisfy the world's needs and wants now.

Computer: Like the One Laptop per Child project?

Dean: That was biased toward what we knew, which is laptops. One Laptop per Child might not be as important as one tablet per child or one cell phone per child. Maybe there's a better model, and we don't need to replicate the laptop for use in the emerging world.

We return to the notion of the keyboard, which is always kind of lingering there. My iPhone and iPad have a keyboard that feels just like the keyboard on my laptop, and I don't have to be tethered to it. So I may be able to break that problem into pieces. There's not a direct link, but some learning should help us focus on current needs.

Forty or 50 years ago, the notion of "if you build it, they will come" worked. Users struggled with all of a device's inefficiencies just because it delivered so much capability. That's not going to happen anymore. Going forward, our guiding principle must be to make computing ubiquitous. I don't want my mother to feel like she's using the computer. I want it to be a part of her normal operating mode, versus adapting to it. And that's the way Watson is going to be delivered.

Computer: What about situations in which people and their device are in one location, but the data is in another location, and the gatekeepers are reluctant to share that data? Healthcare comes to mind, with records locked into incompatible formats from place to place. How do we deal with that? Do you feel the need to evangelize to get the information that was in the PC into whatever device we end up with?

Dean: It will be an opportunity to change the world again. The only way to reduce the rates of illness or death by missed diagnosis or errors is through technology. We've already milked all the process optimizations we can to avoid errors. We need to take the next step, but we need to deliver that capability in a way that's more consumable by doctors, nurses, dentists, and technicians. Patients need to have the authority and feel comfortable saying, "You can have this information, and I trust it will be protected. If I don't want you to have it any more, I can take it back."

We're working on capabilities to make that possible. To make something happen, more people need to say, "Yes, I can" instead of saying, "No, I'm not ready." Unfortunately, it's hard for people to change. Often, these things are generational. Sometimes things happen in places where people are more ready for change to take place.

Initially, I think we'll see the adoption of evidence-based medicine and records in areas where people are ready for this change, particularly where they're more comfortable with socialized medicine. We have to make them comfortable so that we can give them services they would never get without this technology. Doctors need to know that they're going to be able to deliver quality care at a level not possible in the past.

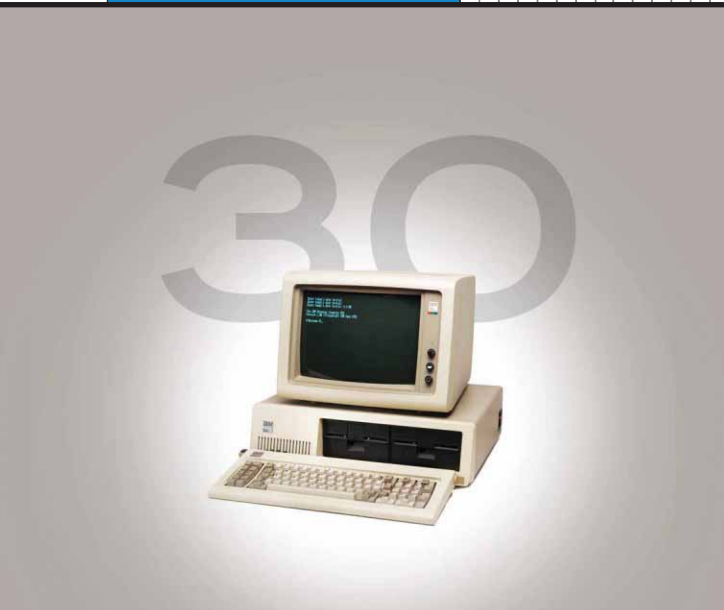
This generational change is beginning to happen. We're tired of the errors, and we want improved healthcare delivery.

Computer: Is the digitization of these common processes one of the legacies of the PC?

Dean: In fact, most systems that control medical devices and interfaces are still DOS-oriented PCs. So they're still catching up to the more modern technology. I think Watson is going to enable the consumability of technology in medicine. We need that.

***Mark Dean**, Chief Technology Officer, IBM Middle East and Africa, is based in Dubai, where he leads IBM's presence in Africa. Dean received a PhD in electrical engineering from Stanford University.*

***Greg Goth** is a freelance technical writer based in Oakville, Connecticut. Contact him at gregorygoth@sbcglobal.net.*



The IBM Personal Computer: A Software-Driven Market

Edward Bride

Instead of designing the fastest or most powerful computing platform based on new technology, IBM chose the fastest path to market, creating its product from already available parts and signing up with a community of independent software vendors to provide the operating system, tools, and a library of applications.

It's common for new technology to be so impressive that it both establishes a market and becomes the standard in that market. More typically, a latecomer may set the standard in terms of market acceptance, even if its newer technology isn't superior. In the consumer video recording area, for example, VHS became the standard, displacing the Beta technology many people considered superior.

Then there's personal computing. Introduced in August 1981, the IBM system wasn't the first personal computer. "Nor was it the most advanced," notes Mark Dean, a member of the original IBM PC design team. But shortly after its launch in 1981, the IBM PC "became the leading platform in the revolution that brought computing out of the glass house and into daily life." (www.ibm.com/ibm100/us/en/icons/personalcomputer)

It's perhaps a historical irony that the brand of microcomputer which became "the standard" for business computing came about by throwing past practice to the winds. Instead of designing the fastest or most powerful computing platform based on new technology, the standard bearer, IBM, chose the fastest path to market, creating its product from already available parts and signing up with a community of independent software vendors (ISVs) to provide the operating system, tools, and a library of applications.

Why did this happen, in the face of IBM's proud tradition of innovation and proprietary technology?

THE PC FRONTIER

As thoroughly documented elsewhere, IBM was under pressure to deliver a product in a year, and that wasn't something likely to be achieved in a lab in a huge company like IBM.

Less than five years before the introduction of the IBM PC, the "microcomputer"—the term personal computer hadn't yet emerged—was something that required either a kit or special skills and training to assemble. Brand names like Altair and Heath weren't uncommon on hobbyists' gift lists, but desktop computers, available from companies like Cromenco or Commodore, were uncommon in business. In some companies, a rare if adventurous businessperson might have an accounting application from Peachtree or Great Plains, running on machines based on the CP/M operating system.

It wasn't IBM that dramatically opened this new frontier, but Apple Computer. Starting in mid-1978, the Apple II computer was often equipped with VisiCalc, a revolutionary business spreadsheet product designed by Dan Bricklin and Bob Frankston, and then available from Personal Software, Dan Fylstra's software publishing company.

 Published by the IEEE Computer Society

VisiCalc changed the way financial analysts operated. I learned of this new tool while working in the marketing department at Hewlett-Packard, in between editorial jobs at *Computerworld* and *Software Magazine*. Instead of working with pencil and paper to develop our annual budgets, we would ask the finance department to use the Apple II and this new type of automated spreadsheet to help make our calculations. Under the watchful eye of the "owner" of VisiCalc, marketing would give and finance would take, but the process still was easier than pre-Apple.

I hadn't read anything about Apple or VisiCalc at *Computerworld*, perhaps largely because the machines were so low in price that they were below the threshold for capital expenses, and weren't visible to either finance or data-processing executives. These desktop computers could be acquired by individuals whose budgets were reviewed at their own departmental level, outside the scrutiny of data-processing departments. Somehow, our finance department knew about this terrific new tool and acquired an Apple II for its own use. The as-yet-unnamed PC market had really been born.

It doesn't matter whether IBM coveted a piece of Apple's pie or simply recognized that the market could be significant for business computing based on the microcomputer platform. These machines were powered by a processor so small that the iconic ex-IBMer Herbert R.J. Grosch once referred to it as "a bump on a wire."

SURVEYING THE MARKETPLACE

By 1980, Apple II and VisiCalc had become hot items, and the market was taking off. It became apparent to IBM that the dominant force in mainframe computing could only become "the IBM" of this market if it launched its product within a ridiculously short timeframe. Around mid-1980, the IBM Corporate Management Committee gave William P. Lowe, systems manager in the Entry Systems Division (part of the General Systems Division) facility at Boca Raton, Florida, one year to accomplish the task. The person who led the charge was Philip D. (Don) Estridge, lab director at the Boca Raton facility.

The timeframe wasn't imposed from out of the blue but actually suggested by Lowe himself. According to the IBM archives:

> … Early studies had concluded that there were not enough applications to justify acceptance on a broad basis and the task force was fighting the idea that things couldn't be done quickly in IBM. One analyst was quoted as saying that "IBM bringing out a personal computer would be like teaching an elephant to tap dance." During a meeting with top executives in New York, Lowe claimed his group could develop a small, new computer within a year. The response: "You're on. Come back in two weeks with a proposal." (www-03.ibm.com/ibm/history/exhibits/pc25/pc25_birth.html)

Burton Grad and Mike Uretsky, who played leading roles in a study that evaluated the business prospects for microcomputers for IBM in 1979, recalled how they came to this position after at least two other "PC" studies. Uretsky, who started the information systems department at New York University and has since retired from NYU's Stern School of Business, was a consultant under contract to IBM. His small consulting company's assignment was to do preliminary market research on possible demand for a small computer. Grad had been an IBMer, and in fact was on the team that helped implement IBM's 1969 "unbundling" of hardware and software pricing. Grad had left IBM and was retained to work on this market research study.

Grad and Uretsky actually conducted two of what could best be called "covert projects" for IBM. Commissioned in 1979 to explore the receptiveness of CIOs, customers, and the third-party ISV community to a new player in the microcomputer market, they were forbidden to disclose who their client was.

It became apparent to IBM that the dominant force in mainframe computing could only become "the IBM" of this market if it launched its product within a ridiculously short timeframe.

"Apple was a toy until VisiCalc came along," Uretsky said. That software product was the real revolution, in the sense that "a really inexpensive program could allow you to do business processes the way you thought about them, the way you told a secretary or an assistant to do some analyses for you," he explained. In the mainframe market, customers purchased computing power as if it were a utility, but "people bought Apple computers so they could get access to the software. It became a software-driven market. That hasn't changed."

Apple and VisiCalc were already well-established when the study was launched, and VisiCalc access was part of the plan from the outset. There would need to be a VisiCalc translator, or a VisiCalc equivalent. "That was a given, a constraint," said Uretsky. "We knew enough to know that that market was software-driven." And, since this would be sold to a business audience, not only would it be software-driven, but "a critical mass would be required to succeed, and that would open up the market for a lot of the smaller companies." The people in IBM felt they had a leg up on recognizing and reaching the market, as the company was already selling office equipment to big and medium-size companies: typewriters, dictating machines, and of course computers; it was thought that this corporate presence would equip IBM's sales force to make bulk sales. "We knew that the credibility of IBM would carry over to any small computer that IBM might want to sell en masse, so that was the most likely path to travel," Uretsky observed.

The numbers regarding the potential market demand were all underestimated, Uretsky recalled, but for a good, or at least an understandable, reason. The consulting agreement's confidentiality clause ensured that no companies being interviewed would know that this was a study being done for IBM. "Apple was out there, and there were some other toys," he noted. CIOs were asked if they would buy a small computer with business software. "There would have been a different response if we had asked that question in the context of the provider being IBM, the company from which people purchased typewriters, dictating machines, punch-card equipment, and other business equipment."

Plain and simple: to be nonproprietary, IBM couldn't control the OS.

The interviewees seemed interested, or at least curious, but there were reservations. For these machines to be viable in a corporate environment, the prospective buyers would need to purchase them in quantities that would drive the machine to become a standard, and that meant in the thousands. To make purchases of that magnitude, the buyer would need confidence that the vendor had a long life expectancy, some longevity, and a current track record.

Since the survey didn't identify IBM, the responses had a very conservative bias, Uretsky indicated. "We knew the bias was there, and in which direction the bias was. One of the internal discussions was: how to present the information to the Corporate Management Committee to go forward, in a mind-set of mainframes." That is, in pulling together reasonable numbers on sales projections, even if the estimates were scaled back, there was still skepticism about any ability to predict strong sales of that many boxes.

"IBM looked at this potential from a technical and business standpoint for at least a three-year period. They were trying to understand what the market was, and what technologies they had that would move them into this area," explained ex-IBMer Grad. "From very early on, all of IBM's hardware and systems software, specifically for the smaller company market with System/3, System/32, and so on, was all based on proprietary technology. The operating system was proprietary; and the applications were developed by IBM: manufacturing control, accounting, everything."

IBM felt that its smaller machine offerings needed to include applications software because the customers wanted to buy solutions for their problems and had little programming expertise, and they had no interest in acquiring it, Grad said.

The covert study team wasn't looking at the Apple II as competition, Grad avers, largely because of that computer's technical limitations. "We were looking at using independent software packages like Peachtree or Great Plains for accounting, which had trouble running on hardware with limitations such as Apple's, with its uppercase alpha characters and limited line size." Venture capitalist Ben Rosen had been pushing Apple because of the availability of VisiCalc, which had, indeed, established that microcomputer as a business machine.

Grad and Uretsky pointed out that this was a new market, not just for IBM but for technology in general, both for individual users and business analysts. They advised IBM not to go proprietary with its systems and applications software because "we knew that in order to succeed, IBM would need a lot of third parties writing software for the new system," Grad recalled. Plain and simple: to be nonproprietary, IBM couldn't control the OS.

CHOOSING THE OS

By running on a standard platform, IBM wouldn't have to fight the battle of convincing third-party application writers to convert to some proprietary IBM system. The 1979 study concluded that essential elements must include a commonly available operating system, preferably already a standard, and that the company would need third parties in that small-systems market. "We recommended that IBM rely heavily on these third parties, and not try to build all the software themselves," Grad said.

IBM agreed that it didn't want a proprietary OS. "The only way to accomplish that was to allow the operating system vendor to put it onto any platform they wanted," Grad noted. "No matter who IBM selected, if it was going to be nonproprietary, it had to be available anywhere.

"That was heresy, to some extent," Grad added. But IBM agreed with this recommendation and sought to find an independent OS.

So, which OS? The choice was clear. It should be CP/M, the established standard from Digital Research Inc. (DRI), previously known by the rather pretentious name Intergalactic Digital Research.

How did IBM end up with Microsoft? There are several versions of how the IBM OS came to be the new DOS instead of the already-established CP/M. (IBM would receive much criticism for its business decision to not make PC-DOS proprietary and to let Microsoft have the right to license MS-DOS to other microcomputer manufacturers, Grad reflected.)

One way or the other, DRI's reticence or unavailability—or whatever the motivation not to do business with IBM—brought the IBM software team, headed by Jack Sams, back to Microsoft.

According to Mark Dean:

> Jack Sams was the engineer in charge of software development for the prototype. He had worked on the IBM System/23 [also known as Datamaster], and had spent a year building the Basic-compiler for it, pushing the product behind schedule. He didn't want to repeat the same struggle with the new microcomputer, so he decided to license most of the software from an outside company. Sams met with Bill Gates to evaluate whether Microsoft could handle the task of writing a Basic compiler for the IBM PC. This led to his recommendation to William Lowe that they use Microsoft software in the final product. In addition, when he was unable to make a deal with Intergalactic Digital Research for the operating system, Sams and his team turned to Microsoft. This led to the development of an operating system released by IBM as PC-DOS and by Microsoft as MS-DOS. (www.ibm.com/ibm100/us/en/icons/personalcomputer/team)

The folklore includes at least three different stories of why DRI didn't get the deal, chronicled in various forms, including Robert X. Cringely's colorful *Accidental Empires: How the Boys of Silicon Valley Make Their Millions, Battle Foreign Competition, and Still Can't Get a Date* (Addison-Wesley, 1992). In Cringely's version, CEO Gary Kildall was flying his plane and didn't or wouldn't come to the meeting; in other versions, lawyers didn't want Kildall to sign a nondisclosure agreement with the big company, or personalities steered them away from doing business with IBM. From a historical viewpoint of assessing the impact of third-party parts, the correct version of the story doesn't matter very much: the key decision was to go to an independent OS provider, rather than which ISV would provide it.

The fact was that IBM was ready to make a deal; they had already signed up with Microsoft for the Basic language, and were ready to make the same sort of arrangement with DRI for CP/M on the new machine. Would DRI have become "a Microsoft," in terms of innovation and industry power? We'll never know for sure, but the point is that to be nonproprietary, IBM couldn't control the OS.

Microsoft had just been formed in 1975, and didn't actually incorporate until June 1981, less than two months before the PC was introduced. Perhaps formally incorporating was a requirement by IBM to continue doing business with these young developers and entrepreneurs.

Once they made that decision, IBM could go to the ISV community and make the promise of offering a nonproprietary platform, mitigating any concerns that IBM might not have the customary market success with microcomputers that it had in mainframes; that is, that it might not actually become "the IBM of the personal computer world."

As the IBM PC developed a larger ISV following in the early 1980s, overcoming the technical deficiencies and limited memory of the initial hardware product became more important. One player not to be forgotten is Lotus Development Corporation, another significant spreadsheet company founded by Mitch Kapor and backed by Ben Rosen. The industry in the 1980s was largely trying to work around the hardware limitations, recalled historian Thomas Haigh, of the University of Wisconsin-Milwaukee. Lotus' Jonathan Sachs originally wrote 1-2-3 in assembly language, Haigh said, giving performance benefits but also resulting in later problems in porting to other platforms.

IBM ANNOUNCES THE PERSONAL COMPUTER

From a 12 August 1981 press release by the IBM Information Systems Division:

Software for Business and Home

"We intend the IBM Personal Computer to be the most useful system of its kind," [C.B. Rogers, Jr., IBM vice president and group executive, General Business Group] said. "Besides making it easy to set up and operate, we are offering a program library that we expect will grow with the creativity of the personal computer users."

Rogers said IBM has established a new Personal Computer Software Publishing Department for the system. It will publish programs written by IBM employees working on their own time and those accepted from independent software companies and outside authors.

Program packages available for the IBM personal computer cover popular business and home applications. For example, EasyWriter will store letters, manuscripts, and other text for editing or rapid reproduction on the printer. Businesses can use General Ledger, Accounts Payable, and Accounts Receivable by Peachtree Software, Inc. to generate balance sheets, track accounts and automatically print checks.

VisiCalc is available for applications ranging from financial analysis to budget planning. Microsoft Adventure brings players into a fantasy world of caves and treasures.

Advanced Operating Systems

IBM, in conjunction with Microsoft, Inc., has adapted an advanced disk operating system to support IBM Personal Computer programs and software development. It has also contracted with Digital Research, Inc. and SofTech Microsystems, Inc. to adapt the popular CP/M-86 and UCSD p-System to the Personal Computer. These two systems should provide users with the opportunity to transfer hundreds of widely used applications to the IBM Personal Computer with minimal modifications.

DOS LIMITATIONS

Because the IBM PC became an industry standard, it shaped the experience of users of its far more powerful machines for decades to come. Overcoming its initial limitations was difficult. Instead, new additions had to be built on top of and around the PC's original architecture. Haigh pointed out that the original extended-memory standard was developed by three companies and nicknamed LIM

Computer Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page Qmags THE WORLD'S NEWSSTAND®

(Lotus-Intel-Microsoft), so Lotus could have a decent-size spreadsheet. IBM eventually acquired Lotus in 1995.

From a computer science standpoint, "DOS was a terrible OS," said Haigh, adding that "it was successful because of its backward compatibility." DOS was originally supposed to be "PC-like" but not totally compatible at the BIOS level, he recalled. By 1994, 32-bit Pentium-based personal computers running Window 3.11 were hundreds of times more powerful than the original PC, but were still limited by its original hardware design. Haigh noted that the design was "far uglier than a clean-sheet alternative would have been," but the machines had evolved to maximize backward compatibility with the original PC running DOS underneath Windows, and relying on a creaky system of segmented memory management. "If there were a properly designed modern OS," said Haigh, "the niche for a product like Quarterdeck Extended Memory Manager (a third-party utility used to optimize free memory within the PC's original 640-KB limit), would never have existed. But with the power of the whole collective bundle of users and standards and providers, everything could move along incrementally, even with the inherent limitations."

A media frenzy ensued after introduction of the IBM 5150, which quickly became known by its more familiar name, the IBM Personal Computer.

Terrible or not, DOS wasn't a very ambitious OS, especially compared to Unix or Mac. In fact, it wasn't even significantly more ambitious than CP/M, commented Haigh. He believes this simple start impacted the structure of the software industry. For example, "WordPerfect's developers had to write their own printer drivers and graphics drivers to support new devices, rather than being able to rely on operating system capabilities." This type of tuning and feature creation made a difference in terms of the kind of resources required to develop an application. "Contrast this with software development effort for today's iPhone or iPad apps, where the OS does so much more for them. Today, there are so many more tools in the OS."

PC MEDIA FRENZY

A media frenzy ensued after introduction of the IBM 5150, which quickly became known by its more familiar name, the IBM Personal Computer. The market took on a consumer persona, spurred by the fact that IBM established its own retail stores and teamed up with both Sears and the ComputerLand chain. Suddenly, IBM PCs were everywhere.

New PC-related publications seemed to debut every week. In one year in the mid-1980s, in fact, 55 new publications came onto the scene; the next year, 55 publications ceased operations—some of them the very same ones that had premiered the previous year. Magazines like *BYTE* and *Creative Computing* played an important role, exchanging information and even printing program listings and circuitry diagrams. Review-oriented media such as *PC Magazine* looked like the Manhattan telephone directory. Editors were so competitive to get their hands on information about new products that they would often review prerelease versions, or in fact write about products that weren't even deliverable. This practice became so common that a name was coined for announced but incomplete software: "vaporware." In one infamous example, a publication declared the multifunction product Ovation to be "product of the year," yet it never even shipped.

Now, more than 25 years later, the term and the practice are memorialized in a Wikipedia entry:

> "Vaporware" was coined by a Microsoft engineer in 1982 to describe the company's Xenix operating system, and first published by computer expert Esther Dyson in 1983. It became popular among writers in the industry as a way to describe products they felt took too long to be released. Vaporware first implied intentional fraud when it was applied to the Ovation office suite in 1983; the suite's demonstration was well-received by the press, but was later revealed to have never existed. (http://en.wikipedia.org/wiki/Vaporware)

(See also E. Bride, "The Media Are the Message: 'The Influencers'," *IEEE Annals of the History of Computing*, Oct.-Dec. 2006, pp. 74-79.)

The staff of one of the mainstream media computer publications spent nearly an entire editorial retreat discussing whether they should even be covering the PC. With its audience being in the IT departments of large corporations, it was thought that the PC wouldn't be a serious contender in enterprises until mainframe connectivity became a reality. Of course, it was just a matter of evolution and time.

Mike Uretsky, having done some of the market research, as well as putting together some of the presentations for IBM's Corporate Management Committee, saw some IBM projections as well. And with regard to the software: "… none of us saw the magnitude of what ultimately evolved. Having said that, we knew that you needed a critical mass of programs, and that they had to be business-related."

EARLY SOFTWARE EFFORTS

Other than VisiCalc, much of the early software effort was "a debacle," recalled Uretsky. "The original attempt at an OS was to port down a mainframe OS, from a mini down to a PC. That was never going to work, for technical reasons such as memory limitations. We would tell people like [IBM vice president of sales and marketing] Buck

Computer | Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page
Qmags THE WORLD'S NEWSSTAND®

Rogers to find a couple of kids with a garage, which indirectly led to a conversation between him and Bill Gates' mother, and the rest becomes history." At least that's one version of history.

About the only proprietary technology in the PC was the BIOS. IBM did this for self-protection, according to Grad. Clones would be sure to form, just as competitive plug-compatible manufacturers did in the mainframe era. The BIOS is central to performance, and keeping it proprietary slowed down the clones' ability to bring a product to market. Eventually, the independents broke through with a BIOS that emulated the IBM PC through reverse engineering, which opened the market to even more manufacturers and further sweetened the market prospects for ISVs.

The BIOS had the IBM hallmark of something proprietary to lock people in. It was still a hardware company then, and it was going to make the money off the machines. They thought that the more they sold, the more they would make, and the more they sold, the bigger the market opportunity for the software companies.

As a lead-in to its willingness to work with ISVs, IBM had already started courting the mainframe software vendors. Recognizing that it couldn't compete in the applications programming business, IBM had convened a "love-in" for ISVs at an industry gathering sponsored jointly with ADAPSO (the Association of Data Processing Service Organizations, later known as the Information Technology Association of America). The message was that IBM welcomed and would work with third-party software developers, and help them succeed. This may have helped to set the stage for IBM to embrace ISVs to provide system and application programs for the IBM PC.

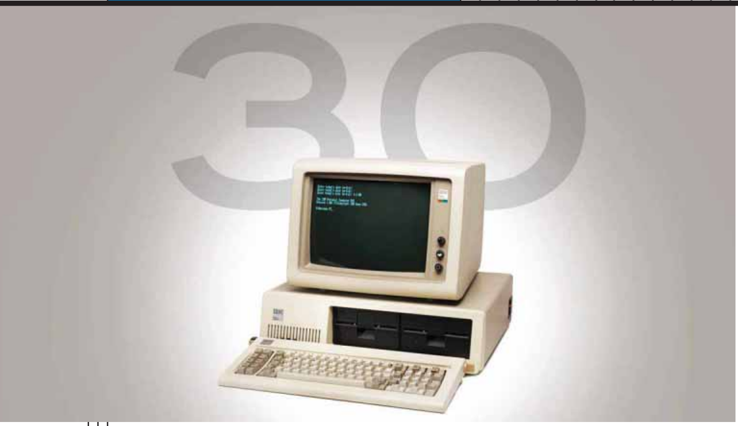
As Grad stated, "The tech-business decisions were interwoven, and were independent of the hardware decisions." Had IBM created its own unique OS, the story would be far different. "The IBM PC shaped the industry. It wouldn't have done so without that open operating system." An educated assessment of what would have happened can be made by looking at the minicomputer world: pre-Unix as a standard operating system, there were a lot of focused successes such as Data General, HP, DEC, and more. All were proprietary, and none experienced the kind of growth the PC later enjoyed. "I believe that the PC market would have grown much more slowly with a closed operating system," said Grad.

IBM's open approach replaced Apple's model (a single platform controlled by one vendor). DOS was the same type of elementary OS as CP/M, although CP/M still required a porting effort to bring applications to different versions. The IBM PC world would be unlike the CP/M world. Proprietariness was the path not taken, and, as Robert Frost said, "That has made all the difference." C

Edward Bride *is an editorial consultant and freelance writer serving the high-tech industry. He was the editor in chief of* Computerworld *and in 1981 became founding editor of* Software News, *later renamed* Software Magazine. *Bride is based in Pittsfield, Massachusetts, where he spends much of his personal time as a member of the boards of four nonprofit organizations that promote jazz education. Contact him at ebride@berkshire.rr.com.*

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

COVER FEATURE



The IBM PC: The Silicon Story

Gurbir Singh

The longevity of the IBM PC architecture, which has had an enormous impact on the evolution of both personal computers and server systems, can be attributed largely to its backward compatibility with existing software applications.

Thirty years ago this month, IBM changed the computing world with the introduction of the personal computer. The IBM PC brought computing to the masses, changing how we work, interact, and entertain ourselves. It profoundly impacted global commerce and communications and was a key element in creating today's interconnected information society.

Even though it was a late entry into the market, the IBM PC was an instant hit. The support of a large, well-respected corporation indicated that this new technology was here to stay. The most cautious customer felt assured that IBM would provide its legendary service, and continued innovation and development in future offerings.

The IBM PC sold around 200,000 units in the first year—no mean feat for a product that was priced higher than other home computers of the time. Sales grew rapidly, and by the mid-1980s it became the unquestioned volume leader. The PC architecture has dominated the industry ever since and is now used in more than 90 percent of all personal computers sold worldwide, generating annual revenues of hundreds of billions of dollars.

EVOLUTION OF PC TECHNOLOGY

IBM created the PC of 1981 with readily available components and opted to use operating system software from outside vendors rather than develop an OS in-house. Rather than overwhelming the buyer with technology innovations, the designers' vision was to create an industrial-strength computer on an exceptionally short development schedule.

The first PC featured the Intel 8088, a 16-bit CPU that was a step above the 8-bit processors in home computers of the day. This processor could address up to 1 megabyte of memory, which would better serve the needs of business applications. The motherboard had a socket for the 8087, the floating-point companion to the 8088. For business users, the PC was configured with a high-resolution, monochrome, text-based video system; two 5-1/4-inch floppy disk drives; and up to 512 Kbytes of memory. The unit shipped with a high-quality, extremely reliable keyboard.

IBM chose the 8088 processor because it was well-supported with a range of peripheral components such as interrupt and direct memory access (DMA) controllers, a system timer, and a bus interface controller. Several of these devices had originally been designed for the 8085, an 8-bit CPU introduced in the late 1970s, and were mature components offered by several manufacturers. The 8088's 8-bit bus was essentially compatible with these devices, giving it a significant advantage over other comparable microprocessors.

The company also chose to clearly document the PC's design by providing the hardware reference manual shown in Figure 1, which included detailed circuit diagrams, a pinout of the connectors for expansion cards, and a listing of the BIOS firmware. IBM encouraged other companies to develop plug-in cards for the PC and expand its uses, and even offered its own prototype adapters. Several companies immediately saw the opportunity and offered cards such as the Hercules Graphics Card, the Creative Labs Sound Blaster audio card, and the 3Com Ethernet adapter.

Other companies opted to copy the entire IBM PC design and offer much cheaper versions of the machine. The inclusion of the Intel 8088 and its support devices in the PC made this decision easier, as all the key components were widely available and the industry had experience with their use. The BIOS firmware was the only portion of the original design copyrighted by IBM. Compaq was

Published by the IEEE Computer Society

Computer Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page

the first company to create a clean-room version of this firmware, and other BIOS vendors quickly followed with their own designs, leading to a significant growth in the number of PC clones. The operating system of choice was PC-DOS, a licensed, rebranded version of Microsoft's MS-DOS. IBM's license to MS-DOS was nonexclusive, and Microsoft sold the operating system to other computer vendors. The various PC clones could all run the software developed for the IBM PC on MS-DOS as long as their system designs were compatible.

IBM nevertheless remained the technology leader, and within four years it introduced several successors to the original PC. The PC XT, and more notably the PC AT, were well-received by customers and became very successful products. The personal computer had graduated from being a toy to become a serious tool for the business user, and the new machines sold by the millions. The PC platform became an accepted standard and attracted even more clone makers. Soon Compaq, Dell, Gateway 2000, and several traditional computer companies such as Digital Equipment Corporation and Hewlett-Packard were shipping PC clones in increasing numbers and challenging IBM in the market.

Figure 1. IBM PC AT technical reference manual and prototype adapter card.

In 1987, IBM launched the Personal System/2 with the proprietary Micro Channel Architecture to reassert its leadership of PC development. The MCA was significantly more advanced than the bus in the PC AT, and IBM was willing to license it to other manufacturers. There were few takers, however, and the 16-bit PC AT remained the standard interface in the vast majority of PCs throughout the 1980s.

As PC volume grew, so did the number and variety of add-in cards to accommodate users' ever-growing and diverse needs. The PC AT bus soon began to show its limitations. Setting up a PC with multiple add-in cards from different vendors became a real challenge: the user had to properly assign I/O ports and memory addresses, interrupts, and DMA channels from a limited set supported by the software drivers for each specific card, at the same time ensuring that these assignments did not collide with those of other cards in the system. At times this proved to be impossible and thus became a serious impediment for the system integrator. In addition, high-performance add-in cards such as display adapters and advanced disk controllers required more bandwidth, and the 16-bit PC AT bus running at 8 MHz constituted a performance bottleneck.

This situation prompted creation of the Peripheral Component Interconnect by an industry consortium in 1993. The PCI interface increased bandwidth eight- to sixteen-fold over the PC AT bus and brought autoconfiguration to the PC. Each card carried device-type information for the software to properly configure the card and select the appropriate driver. A mini-PCI variant was created for use in laptops and other small-form-factor machines.

The PCI bus served the PC's needs for another decade until technological advances overtook its design. Very-high-speed serial interfaces running at several gigatransfers per second became feasible, and PCI Express was launched in 2004. PCIe remains the state of the art today and meets the needs of all PC peripherals.

THE PC'S IMPACT ON SERVERS

The economies of scale afforded by annual high-volume PC shipments began to have a major impact on other areas of computing beyond the personal computer. Compaq was one of the first companies to exploit this emerging ecosystem. It also recognized the Intel 80386 processor's potential and in 1989 launched a line of server systems based on this CPU. These machines captured the price-performance lead as they took advantage of low-cost, mass-produced PC components. Other companies jumped into the market, and soon large numbers of processors were being sold for use in servers, some of which had two or more processors.

Intel saw the opportunities of this new market and chose to cater to its needs. The company took great pains to ensure that its successors to the 80386 were backward compatible with existing software—an indisputable requirement of the PC market to this day. However, the hardware design of system components and interfaces was increasingly tailored to the needs of server systems. Pentium-generation processors could also be connected together to create multiprocessor systems and yet be fully PC-compatible. In the mid-1990s, Intel introduced a range of processor modules for the server market. These modules used the same CPU die designed and developed for the PC but paired it with a high-performance caching system optimized for server workloads. The price-performance ratio of systems built around these modules was unbeatable.

Both AMD and Intel introduced 64-bit extensions to the CPU architecture expressly to benefit the server marketplace. These extensions were made to existing 32-bit x86 processors to take advantage of the development treadmill that is driven by PC market demands and can absorb the high costs of creating new designs. This market also drives innovation and capital investment in silicon process technology and high-volume manufacturing.

As the PC's architecture evolved, it accumulated new features that created numerous compatibility challenges for CPU and system designers.

Codeveloping desktop, laptop, and server processors has its benefits. In each market, the goal is to achieve the highest performance, with the lowest power consumption and the smallest die area, at the lowest cost. However, laptop PCs and servers have the most stringent power-efficiency requirements. Servers impose an additional demand for high reliability, and the common CPU core must be built with more resilient circuits. Numbers of these cores are now married to fast caches, memory controllers, and multiprocessor system interfaces, resulting in CPUs well suited to high-performance server systems. The server processor developers can select the number of cores, the size of the caches, and the capabilities of the memory and system interfaces to offer a range of processors suitable for different sizes and classes of servers. Thanks to the economies created by the PC marketplace, x86-64 processors are now the dominant CPUs in server systems throughout the world.

A COMPATIBLE PLATFORM

The cornerstone of the IBM PC architecture's success is its guarantee of backward compatibility. Customers can depend on future machines to run their (well-behaved) applications, unchanged. Software vendors also benefit from this stability, as they can stretch development efforts over several generations of PCs. Any machine that deviated from the path of compatibility failed in the marketplace. IBM learned this lesson early with the PCjr, a machine that required special versions of software and was never widely accepted. When IBM introduced PS/2 systems in 1987, it made sure these machines were backward compatible with existing software.

This compatibility has been accomplished through consistency in the CPU architecture and platform, BIOS firmware, and OS software and has been driven largely by Intel, BIOS vendors, and Microsoft. Maintaining compatibility over 30 years has been a real challenge. Moreover, escalating complexity makes compatibility a constantly moving target: each step in the evolution of CPUs, the platform, or the software introduces unique compatibility quirks for the next generation. Hence, a company like Intel strives to ensure that its CPU designs—like those shown in Figure 2—meet this requirement. It has achieved this through two main efforts.

First, Intel has created a master reference simulator for all of its processors' instruction sets. The reference simulator was carefully crafted to ensure the utmost architectural accuracy of the output and resulting state of every instruction for all reasonable input conditions. Researchers use this simulator extensively in the design and validation stages for any new CPU to ensure instruction-set compatibility across multiple generations of processors being developed by different teams across the organization. The responsibility of updating the master reference simulator falls on the shoulders of a few experts well versed in the Intel CPU architecture's details.

Second, Intel has developed suites of compatibility tests to ensure that new CPU and chipset designs strictly adhere to PC rules of operation. Ranging from high-level tests for correct operation of interrupts or memory ordering to tests tailored to a specific microarchitecture implementation in a particular generation of CPUs, these tests create the conditions necessary to invoke unusual behavior that may have raised a concern about compatibility in the past. The test suites are well documented so that future design teams knew the purpose of each test.

Intel updates its tests whenever it changes the fundamental nature of a CPU microarchitecture. For example, when the Pentium Pro processor was developed with an out-of-order execution core, a certain number of tests designed for the Pentium with an in-order architecture were no longer useful, as they checked for errors in situations that simply did not exist in the Pentium Pro. The company added a completely new set of tests to the suites to guarantee that new conditions of potential incompatibility were thoroughly tested.

COMPATIBILITY CHALLENGES

As the PC's architecture evolved, it accumulated new features that created numerous compatibility challenges for CPU and system designers.

Big real and virtual 8086 modes

The 8088 processor of the original PC and PC/XT operated without protection or paging within a 1-Mbyte address space, of which the first 640 Kbytes were used by DOS and programs of the day. Several companies developed memory expansion cards with megabytes of

Figure 2. Sample of Intel's IBM PC-compatible processors over three decades. Top row: 8088 to 80486; second row: 80486DX2 to Pentium MMX; third row: Pentium II and Xeon modules, Celeron, and Pentium III; fourth row: Pentium IV to Core i7.

memory that could be accessed through a limited set of address windows in the address space above 640 Kbytes. The memory extenders for DOS became very popular, and many software packages took advantage of this extra memory space to store data and programs or use it as a disk cache.

In 1984, Intel replaced the 8088 with the much faster 80286, and IBM used this processor in the PC AT. The 80286 behaved exactly like an 8088 in real mode and addressed 1 Mbyte of memory. The new CPU introduced memory protection and could address up to 16 Mbytes of memory in protected mode. However, DOS was limited to 640 Kbytes and also could not run in the 80286's protected mode. The PC AT put the memory beyond the 1-Mbyte boundary to good use with memory extenders. The 80286 could quickly transition into protected mode by executing a single instruction but required a reset to reenter real mode. The PC AT incorporated a convoluted circuit to reset the CPU and revert to real mode for compatibility with the 8088.

The following year, Intel released the 80386, a 32-bit processor with memory paging. This CPU had to run all the software developed for the 80286 and the 8088, and it included mechanisms to go back and forth between real and protected modes. The 80386 introduced big real mode as a solution to overcome the performance penalties incurred during these transitions. The new processor also introduced virtual 8086 mode, which allows PC-DOS-compatible software to run in big real mode with 20 bits of addressing. All Intel processors since the 80386 implement big real and virtual 8086 modes as the new base for backward compatibility.

System management mode has been a source of several compatibility issues over the generations of PC processors.

8259A PIC mode

The first IBM PC used one or two programmable interrupt controllers (PICs) to manage interrupts to the CPU. Intel developed the 8259A PIC for the 8085 processor and later extended it to work with the 8086/8088 family in 1979. The original PC had one such controller to field seven interrupts, while the CP/AT used two to handle up to 15 interrupts. Much of the software developed for PCs handled its own interrupts and communicated with the PICs. The 8259A was effective in single-processor systems once a designer learned to handle all its quirks.

In 1992, Intel introduced the advanced PIC architecture to support multiprocessor systems. However, the APIC architecture had to incorporate a mode of operation that mimics the 8259A's behavior and exists in every APIC to this day. This mode requires that the device translate an interrupt into a message on the APIC bus and send it to the designated processor, which then completes the handshake of the end-of-interrupt sequence required by the 8259A. Throughout the 1990s, all PCs used this mode of operation while running Windows. Windows XP, introduced in 2001, used the 8259A mode of operation in single-processor systems.

Instruction set

PC processors have evolved at a rapid pace during the past 30 years. Intel and AMD have added several new groups of instructions to the complement recognized by the 8088. To maintain backward compatibility, however, not a single instruction has been dropped from the original set. Hence, every CPU die carries hardware for all instructions, some of which modern software seldom uses. This hardware exacts a penalty, as the CPU consumes more power and it becomes increasingly expensive to test every instruction for correct operation.

Processor architects thus face a major dilemma when working on any new CPU implementation. To achieve the highest overall performance with the lowest power consumption and smallest die size, they must trade off the performance of certain groups of less commonly used instructions while tuning the performance of those instructions that will be used most frequently. Architects must strike the right balance, as such tradeoffs will impact the huge base of existing software. They also must anticipate the future needs of software and operating systems because the design cycles of new CPU implementations can be as long as four to five years.

System management mode

In the late 1980s, demand for notebook computers grew rapidly. The IBM PC and the PC-DOS software it ran were designed for desktop systems, in which power consumption was not a primary concern. The original PC used large numbers of power-hungry bipolar logic devices, and the OS and system firmware had no provisions to power down portions of the system not in use. Considerable innovation in power management and low-power design was required before the PC-compatible laptop could run on batteries for more than a few hours, let alone all day.

To address this problem, Intel introduced system management mode in the 80386SL processor designed for notebooks and other power-critical systems. With SMM, system manufacturers could add power-management features with minimal hardware and extensions to the BIOS. Its primary purpose was to offer a way to suspend the system and put it in a low-power mode during times of inactivity. SMM is entirely transparent to any OS. The SMM code runs in a portion of system memory that has been locked at boot time and is unavailable to the OS. The system hardware can invoke the code in this portion of memory at any time by sending a dedicated system management interrupt signal. Upon receipt of this interrupt, the processor saves its state and starts executing the SMM code in the locked address space.

One of the first uses of SMM was to reduce power consumption by a PC while idle at a DOS prompt. The OS continuously polled the system for any input, keeping all hardware fully active. A HALT instruction was placed in the polling loop by patching the code, and it suspended CPU operation and significantly lowered the processor's power state. A timer regularly triggered SMM, and the SMM handler would step the CPU past the HALT, allowing the processor to proceed around the polling loop at a far lower slower rate than the original power-hungry spin.

SMM enabled manufacturers to add unique features that distinguished their systems from identical clones, and they soon put it to creative uses both good and bad. Today, SMM together with the advanced power-down modes of the processor and support devices are at the heart of all laptop, desktop, and server power-management mechanisms.

This mode has served other needs: it offers most of the capabilities necessary to create powerful debugging tools entirely independent of the OS. SMM-based debuggers can home in on several types of hard-to-find bugs by using code run in SMM to patch in the right results in the CPU registers, under the covers, and then let the program proceed.

SMM has been a source of several compatibility issues over the generations of PC processors. Because SMM is a completely independent mode of operation, its interactions with other CPU modes, interrupts, and faults have changed over time. For example, SMM interaction with the APIC and nonmasked interrupts was inconsistent between generations of CPUs. Fortunately, programmers have created CPU-specific code to handle these inconsistencies.

Multiprocessor support

The Pentium processor family brought multiprocessing to the x86 architecture and raised several backward-compatibility concerns.

Memory ordering was one significant issue. Programs for the PC assume that a sequence of write operations to system memory will complete in exactly the same order as laid out in the program. Single-CPU systems built around a processor with a well-ordered execution pipeline exhibit this behavior, and it became a compatibility requirement. The out-of-order architecture of the Pentium Pro developed for multiprocessor systems brought this issue to the forefront. This processor got a significant boost in performance by executing many independent instructions in parallel. A sequence of independent writes might not complete in exactly the same order as specified in the program and would thus violate the compatibility rules. This problem was solved by creating an ordering boundary within the processor's microarchitecture to contain all unordered operations and transactions. All memory operations would follow strict write ordering from that processor once transactions crossed this ordering boundary, thereby ensuring compatibility.

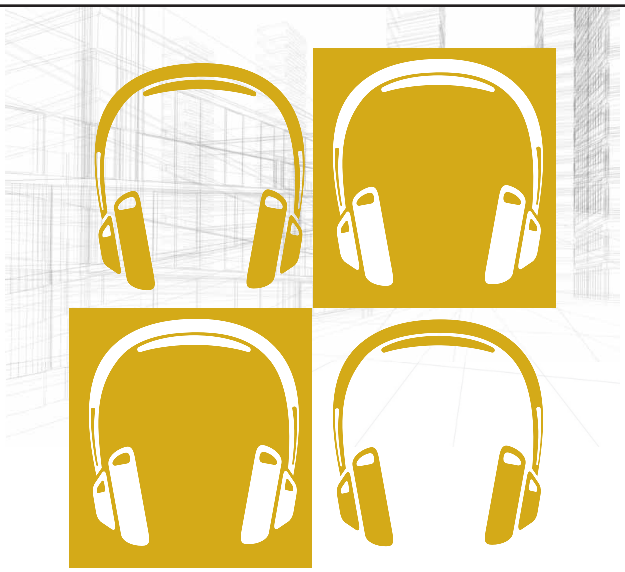
Another compatibility-related concern is that the x86 instruction set supports operations performed directly on operands stored in system memory. These instructions lock the entire system memory while the operands are fetched, operated on, and written back to memory to ensure atomicity during the operation. This behavior has significant performance penalties in a multiprocessor system. Researchers have developed a mechanism to lock the specific addresses in the processor's cache and complete the operations atomically without tying up the entire multiprocessor system's memory. This lock-caching mechanism is robust enough to handle misaligned data types that could extend across cache-line boundaries without creating a deadlock while trying to lock both cache lines.

The IBM PC architecture launched three decades ago is as strong as ever, and has evolved over the years into a robust, capable design that holds the unquestioned lead in the personal-computer marketplace. When the Moore's law engine finally sputters to a halt, we may not know what technology will replace silicon, but we can be reasonably confident that, one way or another, future generations of computers will be able to run legacy code. ■

Acknowledgments

Thanks to Keshavan Tiruvallur, Fellow at Intel Corporation, and Robert Colwell, currently with DARPA and lead architect on the Pentium Pro, for their valuable input and constructive comments that have made this article possible.

Gurbir Singh *is an electrical engineer who recently retired from Intel as a senior principal engineer. He joined Intel in 1984 and worked on the 80960, Pentium Pro, Pentium II, Pentium III, Pentium 4, and Core i7 processors. Singh has 29 patents related to his research on multiprocessor systems, buses, and caches. He received an MSc in electrical engineering from Clemson University and a BTech in electronics engineering from the Indian Institute of Technology, Kharagpur, India. When not tinkering in his machine shop, Singh can be found in his observatory, stargazing. Contact him at gbsingh175@gmail.com.*



Remote Display Solutions for Mobile Cloud Computing

Pieter Simoens, Filip De Turck, Bart Dhoedt, and Piet Demeester, ***Ghent University***

Proposed optimization techniques address the major challenges that varying wireless channel conditions, short battery lifetime, and interaction latency pose for the remote display of cloud applications on mobile devices.

Mobile devices have become an essential part of our daily life, with smartphone sales now surpassing desktop system sales.[1] As mobile device popularity grows, end-user demands to run heavier applications are also increasing.

Although advances in miniaturization continue, the desire to preserve the advantages mobile devices have over desktop systems in weight, size, and device autonomy will always impose intrinsic limits on processing power, storage capacity, battery lifetime, and display size. Researchers must redesign conventional desktop applications to operate on mobile hardware platforms, thereby often reducing functionality, whereas more demanding applications typically require specific hardware resources that are unlikely to be available on mobile devices.

At the same time, the Web hosts increasingly powerful computing resources, offering applications ranging from simple word processors to all-encompassing enterprise resource planning suites to 3D games.[2,3] Both Microsoft and Google have developed complete online office suites—Office Live and Google Apps, respectively—that may emerge as all-round alternatives to their desktop office suites.

Cloud computing can broaden the range of applications available to mobile users beyond conventional office applications by supporting applications requiring graphical hardware, such as 3D virtual environments, or large storage capacity, such as 3D medical imaging. As multiple users share the cloud infrastructure, this technology can provide these hardware resources cost-effectively.

Mobile cloud computing offers a way to meet users' increasing functionality demands, as distant servers execute all application logic and only user interface functionalities reside on the mobile device. The device acts as a remote display, capturing user input and rendering display updates received from the server. However, short battery lifetime, varying wireless channel conditions, and interaction latency present major challenges for the remote display of cloud applications on mobile devices. Researchers have recently proposed several solutions to these issues.

MOBILE CLOUD COMPUTING CHALLENGES

Essentially, mobile cloud computing physically separates the user interface from the application logic. The mobile device executes only a viewer component, operating as a remote display for the applications running on distant servers in the cloud. As Figure 1 shows, a remote display framework has three components: a server-side component that intercepts, encodes, and transmits the application graphics to the client; a viewer component on the client; and a remote display protocol that transfers display updates and user events between both endpoints.

Using standard thin-client solutions such as Microsoft's Remote Desktop Protocol (RDP), Citrix Systems' Independent Computing Architecture (ICA), and AT&T's Virtual Network Computing (VNC) in a mobile cloud computing context is not straightforward. These architectures were originally designed for corporate environments, where

Published by the IEEE Computer Society

users connect over a wired local area network (LAN) to the central company server executing typical office applications. In this setting, the technical challenges are limited because delay and jitter are minimized, bandwidth availability is seldom a bottleneck, and office applications exhibit rather static displays compared to multimedia applications. In a mobile cloud computing environment, the remote display protocol must deliver complex multimedia graphics over wireless links and render these graphics on a resource-constrained mobile device.

A potential obstacle to mobile cloud computing success is the encumbered I/O functionality of mobile devices. Slideout keyboards and stylus devices facilitate user input and maximize display sizes without increasing overall device size, but they do not provide an adequate solution for convenient I/O. Novel user interfaces for mobile devices are, however, entering the market. Media tablets, such as Apple's iPad, offer a larger screen with touch functionality and a keyboard close in size to regular keyboards. Other manufacturers, such as NEC, minimize their device size and support external keyboards and displays[4] to augment I/O functionality.

More fundamental challenges for mobile cloud computing lie in the short battery lifetime of mobile devices, the limited and varying bandwidth on wireless links, and the interaction latency between user input and display updates. Table 1 presents an overview of several solutions proposed to address these challenges.

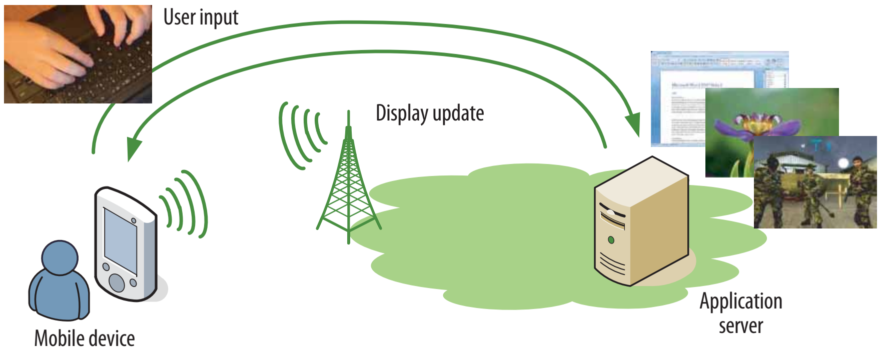


Figure 1. Remote display framework. The viewer component on the client forwards captured user input to the server. In turn, the server-side component intercepts, encodes, and transmits application output. A remote display protocol transfers display updates and user events between both endpoints.

Table 1. Proposed solutions to mobile cloud computing challenges.

Challenge	Solutions
Device battery lifetime	Cross-layer identification of wireless network interface card (WNIC) sleep intervals
Wireless bandwidth availability	Versatile graphics encoding Downstream data peak reduction Optimization of upstream packetization overhead
Interaction latency	Proximate cloudlet infrastructure Computing display updates in advance Image buffering for virtual environment streaming Scene object caching

DEVICE BATTERY LIFETIME

The operational time of mobile devices is often limited when they are extensively used. These battery capacity shortcomings result in short recharge cycles and prevent users from relying completely on their mobile device. During the past decade, advances in nominal battery capacity have been modest. Kostas Pentikousis[5] observed that technological improvements are currently stagnating because of the lack of a major battery technology breakthrough comparable to the advent of rechargeable Li-ion batteries. Consequently, extending device autonomy should primarily be realized by making the device itself more energy efficient.

Offloading mobile applications

At first glance, offloading mobile applications to the cloud appears to be a straightforward way to save on device energy consumption because it reduces the amount of local processing. As Figure 2 shows, however, there is a tradeoff between local processing and network bandwidth consumption, and bidirectional communication with the application server incurs additional energy drain from the battery via the wireless network interface card (WNIC).

Karthik Kumar and Yung-Hsiang Lu[6] modeled this tradeoff and concluded that, from an energy perspective, offloading applications from mobile devices is mainly advantageous when large amounts of computation are needed in combination with relatively small amounts of network communication. Demanding applications exchange a significant quantity of data between client and server because they entail a high degree of interactivity and detailed graphics—for example, walking around in a 3D virtual environment or rotating 3D medical images. Thus, according to Kumar and Lu's model, offloading such applications is not beneficial. On the other hand, demanding applications have important

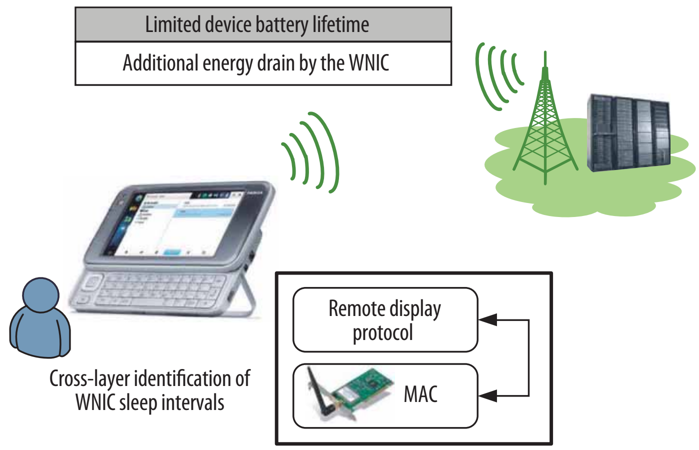


Figure 2. Offloading mobile applications to the cloud reduces the amount of local processing, but there is a tradeoff because bidirectional communication with the application server incurs additional drain from the battery via the wireless network interface card (WNIC). A proposed cross-layer power-saving approach operates between the media access control (MAC) layer and the remote display protocol layer.

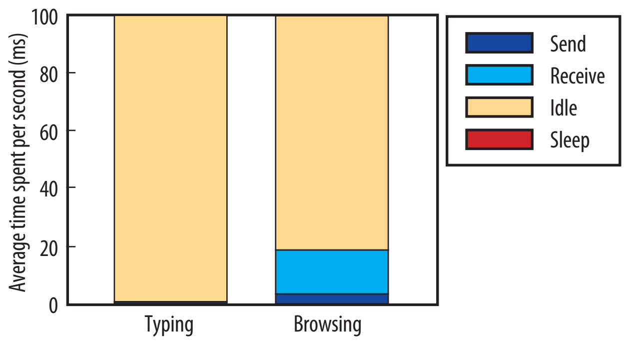


Figure 3. Average time distribution of the four possible WNIC modes—send, receive, idle, and sleep—resulting from remote display protocol traffic in two different scenarios: typing and browsing. The WNIC is mainly in idle mode and seldom enters sleep mode.

Table 2. Cisco Aironet WNIC power consumption.

Power mode	Measured power (mW)
Send	1,090-1,550
Receive	1,060-1,380
Idle	1,150
Sleep	300

hardware requirements that typically can only be met in the cloud and therefore need to be offloaded or remain unavailable to mobile users.

Cross-layer identification of WNIC sleep intervals

To develop strategies that optimize the energy balance, it is important to study WNIC energy consumption, which is the product of the number of bytes exchanged over the wireless interface and the energy cost per byte. The average energy cost per byte is determined by the distribution of time over the four possible WNIC states: send, receive, idle, and sleep. Because a specific set of WNIC components are activated in each state, power consumption varies widely between the states. Figure 3 illustrates average WNIC time distribution in two typical remote display scenarios, typing and browsing, while Table 2 lists the power consumption in each state for a typical WNIC, the Cisco Aironet.[7]

Although the send and receive modes consume the most power, energy-saving approaches should focus on the large idle times observed in remote display scenarios. These idle times are a consequence of the limited frequency of user interactions imposed by the network round-trip time. After some interaction, users must wait until the results become visible on the screen before continuing their work. Furthermore, interactive applications will only update their display when instructed by the user—for example, by entering a URL or clicking on a hyperlink.

Major energy savings are expected when the WNIC transitions to energy-conserving sleep mode during these idle intervals. Sleep mode consumes three to five times less energy than idle mode because the radio interface is turned off. Of course, this implies that the WNIC will miss any incoming data when it is in sleep mode, so the sleep intervals must be carefully chosen.

A proposed cross-layer power-saving approach operates between the MAC layer and the remote display protocol layer (see Figure 2).[8] Because the MAC layer operates on binary data and cannot discriminate between, for example, transmitted user input and Transmission Control Protocol (TCP) acknowledgments, it is unaware of the arrival of the next display update. The appropriate sleep intervals must therefore be determined at the remote display protocol layer, where the display update schedule is established—for example, via a push approach in which the server sends display updates with fixed intervals or a pull approach in which the client sends an explicit request. Correlating the transmission of user input to the network round-trip time predicts the arrival of the next display update. In between two display updates, the WNIC enters sleep mode. This sleep mode is interrupted at regular intervals to transmit user events. Researchers have used cross-layer optimization to reduce WNIC energy consumption by up to 52 percent.[8]

WIRELESS BANDWIDTH AVAILABILITY

Compared to fixed access networks, modern broadband mobile and wireless technologies offer limited and variable bandwidth availability. Universal Mobile Telecommunications System (UMTS) users typically receive up to 384 kilobits per second, while Krishna Balachandran and colleagues[9] reported practical throughputs of 347 Kbps for Long Term Evolution (LTE) and up to 6.1 Mbps for WiMAX. Actual throughput depends on user mobility, interference, and fading effects.

Beyond the higher costs associated with limited broadband availability, other economic factors are motivating the development of highly efficient remote display compression technologies. Mobile users are increasingly confronted with volume-based subscription plans—such as that adopted by AT&T in 2010—and these users will not tolerate the sending of redundant bytes on the network. In fact, the lack of such technologies could impede development of new cloud applications.[10]

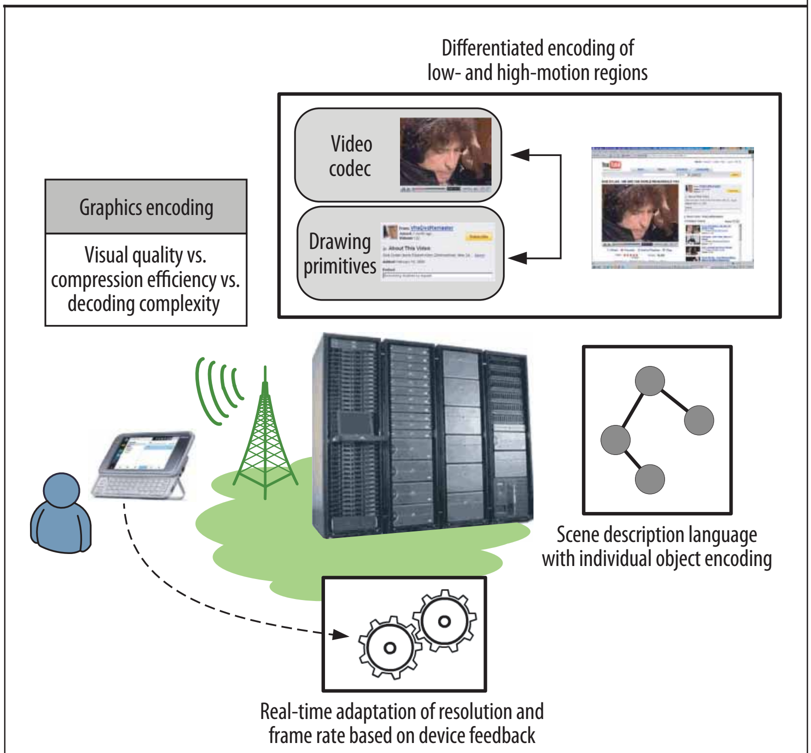


Figure 4. The choice of codec to compress the intercepted application graphics at the server is a tradeoff between visual quality, compression efficiency, and decoding complexity. Remote display updates are optimally encoded through drawing primitives or video codecs, depending on the graphics characteristics. Low- and high-motion scenes must be encoded separately, requiring runtime detection algorithms or the use of semantic scene description languages. Video streaming parameters must be adapted to device feedback.

Versatile graphics encoding

As Figure 4 shows, the choice of codec to compress the intercepted application graphics at the server is a tradeoff between visual quality, compression efficiency, and decoding complexity.

Conventional remote display architectures including RDP, ICA, and VNC typically virtualize a layer of the graphic rendering stack at the server and forward intercepted drawing primitives to the client, such as instructions to draw a rectangle, display a bitmap, or put some text on the screen. This approach is optimal for applications, such as typical office applications, that only update small regions of the display or have a slow refresh rate with respect to the network round-trip time. Bandwidth requirements to remotely display this type of graphics do not exceed 200 Kbps and can be adequately served over wireless links.

On the other hand, encoding multimedia graphics applications would require numerous drawing primitives because they update large parts of the screen at high refresh rates and they often contain fine-grained and complex color patterns. This kind of graphics can be more efficiently encoded using a video codec, such as H.264 or MPEG-4. Using video codecs for remote display purposes is referred to as *interactive live streaming* because the graphics are mainly the result of user interaction, in contrast to regular video streaming, which requires only limited user interaction—for example, to start and stop the video. Researchers have successfully applied interactive live streaming in remote 3D virtual environments[2] and games.[11]

Even when only a single application is used, the characteristics of the graphics on the user display might significantly differ when a user is accessing mobile cloud computing services. For example, a user browsing a Wikipedia page might click on a link that opens a YouTube video in the same browser window. Remote display frameworks must therefore be able to switch seamlessly between multiple encoding modes based on an analysis of graphics at the server (see Figure 4).

Kheng-Joo Tan and colleagues[12] compared the pixels of subsequent frames and split each individual frame into low- and high-motion regions, respectively, encoding them through drawing commands or as H.264 video frames. This hybrid approach operates at the pixel level, which offers the advantage of cross-system applicability because it is the lowest layer of the rendering stack. Transparency at the pixel level, however, comes at the expense of losing any information on the nature of each object in the scene. Consequently, this technique uses the same encoding format

Computer | Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page

Table 3. Packetization overhead of TCP/IP headers when sending a single keystroke to the server.

Protocol	Payload (bytes)	Overhead (percent)
VNC Remote Framebuffer Protocol	8	83.33
Remote Desktop Protocol	6	86.96
Independent Computing Architecture	6	86.96

for low-motion regions regardless of whether they contain text characters or images.

Operating at a higher level in the rendering stack, Mihai Mitrea and colleagues[13] intercepted high-level X11 drawing commands and encoded them using the MPEG-4 Binary Format for Scenes (BiFS), a powerful scene description language. Based on the intercepted X11 commands, they constructed an internal scene graph and converted it to BiFS semantics. A distinctive feature of BiFS is that this scene graph, containing the nature of each object in the scene, is binary encoded and streamed to the client. This allows encoding each object in its own optimal scheme.

While the characteristics of the application graphics mainly determine the choice of encoding format, the actual encoding parameters must be dynamically adapted to cope with wireless bandwidth fluctuations and heterogeneous mobile device capabilities. Numerous factors impact wireless link quality, including device antenna technology, distance from the access point, user speed and fading, and interference effects. At the same time, the various hardware configurations of commercial mobile devices induce variations in decoding power.

Gianluca Paravati and colleagues[14] developed a closed-loop controller for interactive live streaming that optimizes the settings of the video codec parameters based on feedback from the client device. The mobile device regularly reports on the amount of data that the controller encodes per unit of time, a metric reflecting both the device hardware capabilities and the amount of data the device receives. By adjusting the resolution and image quality accordingly, the controller maintains a target frame rate that ensures a smooth visualization experience.

Downstream data peak reduction

Interactive applications only update their display when instructed by the user. These display updates usually involve sending a large amount of data to the client in a short interval, which requires an instantaneous bandwidth much higher than the average bandwidth requirement. Furthermore, this bursty traffic pattern is unfavorable in wireless network environments, as it might induce additional collisions on the wireless channel.

A study of this problem by Yang Sun and Teng-Tiow Tay[15] revealed that repainting the graphical objects after recurring user actions causes considerable redundancy in remote display protocol traffic. Their proposed hybrid cache-compression scheme uses the cached data as history to better compress recurrent screen updates, with the cache containing various drawing orders and bitmaps. Using RDP, they reduced the number of data spikes by 27 to 42 percent, depending on cache size, resulting in global network traffic reductions of 10 to 21 percent.

Optimization of upstream packetization overhead

User events are the principal source of upstream remote display traffic from client to server. Individually, each user event embodies only a small amount of information: a key or button ID, one bit to discriminate between the press and release actions, and possibly the current pointer coordinates. Nevertheless, user events induce important upstream traffic because they are often generated shortly after each other. Entering a single character results in two user events to indicate the press and release actions, whereas moving the mouse results in a sequence of pointer position updates.

Usually, the system transmits user events as they occur to minimize interaction latency. Because data packets sent upstream often contain a single user event, headers added at the TCP, IP, and wireless link layer cause large packetization overhead. Table 3 quantifies the packetization overhead of TCP/IP headers of three commonly used remote display protocols—the VNC Remote Framebuffer (RFB) Protocol, RDP, and ICA—when sending a single keystroke to the server. Optional headers and the wireless link layer header further increase total overhead.

Buffering user events at the client for a short period enables the joint transmission of multiple user events. The maximum buffering period is based on a tradeoff of remote display bandwidth reduction and interaction latency. A closed-loop controller running at the client integrates the interaction latency models in terms of this buffering period and the network round-trip time, ensuring that the average interaction latency does not exceed a predefined maximum value by continuously monitoring the current network status and adjusting the buffering period accordingly.[16] The highest bandwidth reductions are achieved for interactive applications with frequent user events and lower round-trip times. For a text-editing scenario and network round-trip times less than 50 ms, researchers have achieved bandwidth reductions up to 78 percent.

INTERACTION LATENCY

While technological advances are likely to overcome bandwidth limitations, interaction latency—the delay users experience between generating some input and seeing the result on their display—is an intrinsic challenge of mobile cloud computing because the device must communicate even the most trivial user operations to the server.

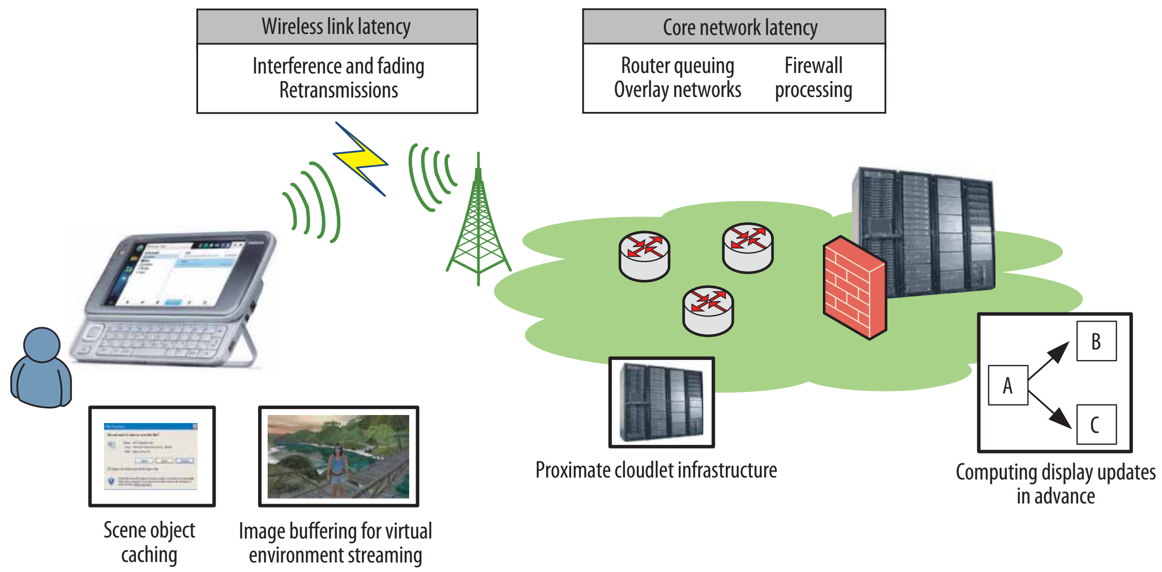


Figure 5. Strategies to mitigate interaction latency focus either on reducing the propagation delay by deploying the application on the proximate infrastructure or on reducing the synchronization between client and server by caching key objects or frames.

Niraj Tolia, David Andersen, and M. Satyanarayanan[17] pointed out that these trivial interactions, such as moving the pointer to draw a line or to select some text, are far more problematic than loosely coupled tasks, such as Web browsing. The major difference is that users expect an immediate visual result from trivial operations, whereas they anticipate processing and download delays when clicking on a link.

Accustomed to the responsive interfaces of desktop applications, users will expect the same interactivity in a mobile cloud computing setting. As Figure 5 shows, remote display protocol data needs to traverse numerous links, both wireless and wired, and multiple network elements, each introducing additional propagation and transmission delays on the end-to-end path. Loss-correcting retransmissions on the wireless link, router queuing, suboptimal routing schemes, and firewall processing all entail significant propagation delays. Bandwidth limitations on the wireless link induce additional transmission delays, especially for immersive applications such as virtual environments that transfer highly detailed graphics from the server to the client. Sometimes several client-server interactions are required before a display update can appear on the screen—for example, when the server waits for the client's acknowledgment before sending the remainder of the data.

Solutions to mitigate interaction latency try to either reduce the number of hops on the end-to-end path by moving the application closer to the client or provide better synchronization mechanisms between client and server (see Figure 5).

Proximate cloudlet infrastructure

Satyanarayanan and colleagues[18] introduced the concept of *cloudlets*: trusted, resource-rich computers dispersed over the Internet. Exploiting virtual machine technology, mobile devices rapidly deploy their services on the closest cloudlet by uploading an overlay VM to customize one of the generic base VMs commonly available on all cloudlets. The physical proximity ensures low-latency, one-hop, high-bandwidth wireless LAN access—for example, over the latest Wi-Fi 802.11n technology—instead of mobile radio technology access, such as high-speed downlink packet access (HSDPA) or LTE.

Despite its promise, the cloudlet concept may require the transfer of data from the central application server to nearby public infrastructure, which can be undesirable for security or privacy reasons. In these cases, latency optimization strategies must focus on reducing the number of round-trip times required to resynchronize the client device display with the server.

Computing display updates in advance

Given the current application state, the application server can predict potential display updates and stream them in advance to the client. Contrary to video streaming, in which the frame order is known in advance, in mobile cloud computing, the next display update depends on user input. For example, when a user opens an application menu, the server can precompute all dialog windows that can be opened by selecting one of the menu items.

Richard Pazzi, Azzedine Boukerche, and Tingxue Huang[19] applied this approach to virtual 3D environ-

Computer Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page
Qmags THE WORLD'S NEWSSTAND®

ments. Given the current user position, the system calculates the possible next user viewpoints in advance and provides them to the client. When the user actually moves forward, the client fetches the correct viewpoint from its cache.

Image buffering for virtual environment streaming

Due to limitations in mobile bandwidth and mobile device memory resources, in most cases, streaming all possible next display updates in advance is unfeasible. Furthermore, the gains of this precomputing technique are highly dependent on prediction accuracy. A better strategy might be to buffer some key display updates, for which the server only needs to provide a differential update. Boukerche, Pazzi, and Jing Feng[20] evaluated several cache management strategies and reduced the amount of requests during a 300-step movement in a 3D virtual environment from 300 to 145. Of course, in this case, the server response is still required to update the display.

Scene object caching

For more static applications, such as office applications, the potential next updates can be more accurately predicted as, for example, a menu layout will rarely change. Consequently, the number of corrective server updates will be more limited. A typical example would be the list of recently opened files in a text editor's File menu. Scene description languages such as MPEG-4 BiFS are particularly suited to support this client-side handling of user input.[13] The client not only receives graphic updates, but also is informed about the structure of the displayed scene and its composing objects, as well as how the user can manipulate these objects.

By physically separating the user interface from the application logic, mobile cloud computing allows access to even the most demanding applications from intrinsically resource-constrained mobile devices. Developers tailor contemporary remote display optimization techniques to mobile devices' short battery lifetime, the varying and limited bandwidth availability on wireless links, and interaction latency. Although each of these solutions adequately addresses specific mobile cloud computing challenges, an overall approach is currently lacking.

Because of user mobility, the wide diversity of applications, and the varying wireless channel status, the mobile cloud computing context is highly dynamic. Future research should therefore focus on the design of a comprehensive framework that integrates the existing solutions and activates the most appropriate one depending on the current device, network, and cloud server status. ■

References

1. C. Albanesius, "Smartphone Shipments Surpass PC Shipments for First Time. What's Next?" *PC Magazine*, 8 Feb. 2011; www.pcmag.com/article2/0,2817,2379665,00.asp.
2. V.S. Pendyala and S.S.Y. Shim, "The Web as the Ubiquitous Computer," *Computer*, Sept. 2009, pp. 90-92.
3. F. Lamberti and A. Sanna, "A Streaming-Based Solution for Remote Visualization of 3D Graphics on Mobile Devices," *IEEE Trans. Visualization and Computer Graphics*, Mar./Apr. 2007, pp. 247-260.
4. H. Kawashima et al., "Virtual PC-Type Thin Client System," *NEC Technical J.*, Sept. 2007, pp. 42-47.
5. K. Pentikousis, "In Search of Energy-Efficient Mobile Networking," *IEEE Comm. Magazine*, Jan. 2010, pp. 95-103.
6. K. Kumar and Y.-H. Lu, "Cloud Computing for Mobile Users: Can Offloading Computation Save Energy?" *Computer*, Apr. 2010, pp. 51-56.
7. T. Kim et al., "An Energy-Aware Transmission Mechanism for WiFi-Based Mobile Devices Handling Upload TCP Traffic," *Int'l J. Comm. Systems*, May 2009, pp. 625-640.
8. P. Simoens et al., "Cross-Layer Optimization of Radio Sleep Intervals to Increase Thin Client Energy Efficiency," *IEEE Comm. Letters*, Dec. 2010, pp. 1095-1097.
9. K. Balachandran et al., "Performance Assessment of Next-Generation Wireless Mobile Systems," *Bell Labs Technical J.*, Feb. 2009, pp. 35-58.
10. C.C. Miller and B. Stone, "App Makers Worry as Data Plans Are Capped," *The New York Times*, 6 June 2010; www.nytimes.com/2010/06/07/technology/07data.html.
11. I. Nave et al., "Games@Large Graphics Streaming Architecture," *Proc. IEEE Int'l Symp. Consumer Electronics* (ICSE 08), IEEE Press, 2008, pp. 205-208.
12. K.-J. Tan et al., "A Remote Thin Client System for Real-Time Multimedia Streaming over VNC," *Proc. IEEE Int'l Conf. Multimedia and Expo* (ICME 10), IEEE Press, 2010, pp. 992-997.
13. M. Mitrea et al., "BiFS-Based Approaches to Remote Display for Mobile Thin Clients," *Proc. SPIE*, vol. 7444, 2009, p. 74440F, doi:10.1117/12.828152.
14. G. Paravati et al., "A Feedback-Based Control Technique for Interactive Live Streaming Systems to Mobile Devices," *IEEE Trans. Consumer Electronics*, Feb. 2010, pp. 190-197.
15. Y. Sun and T.T. Tay, "Analysis and Reduction of Data Spikes in Thin Client Computing," *J. Parallel and Distributed Computing*, Nov. 2008, pp. 1463-1472.
16. P. Simoens et al., "Upstream Bandwidth Optimization of Thin Client Protocols through Latency-Aware Adaptive User Event Buffering," *Int'l J. Communication Systems*, May 2011, pp. 666-690.

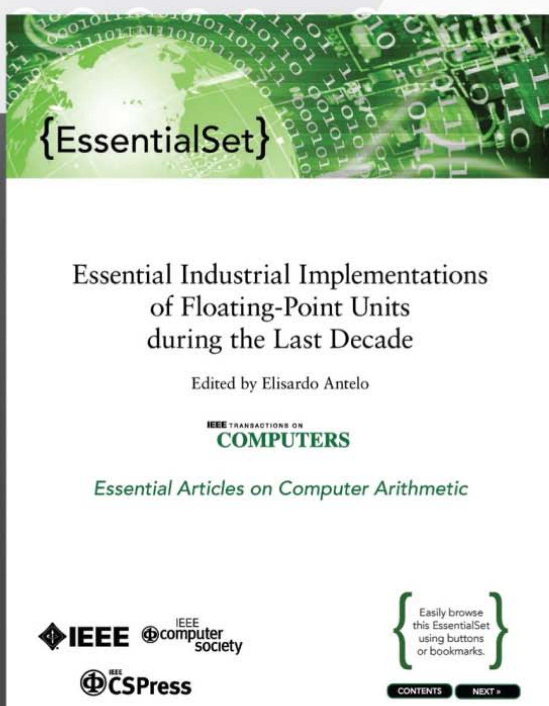
17. N. Tolia, D.G. Andersen, and M. Satyanarayanan, "Quantifying Interactive User Experience on Thin Clients," *Computer*, Mar. 2006, pp. 46-52.
18. M. Satyanarayanan et al., "The Case for VM-Based Cloudlets in Mobile Computing," *IEEE Pervasive Computing*, Oct.-Dec. 2009, pp. 14-23.
19. R.W.N. Pazzi, A. Boukerche, and T. Huang, "Implementation, Measurement, and Analysis of an Image-Based Virtual Environment Streaming Protocol for Wireless Mobile Devices," *IEEE Trans. Instrumentation and Measurement*, Sept. 2008, pp. 1894-1907.
20. A. Boukerche, R.W.N. Pazzi, and J. Feng, "An End-to-End Virtual Environment Streaming Technique for Thin Mobile Devices over Heterogeneous Networks," *Computer Comm.*, July 2008, pp. 2716-2725.

Pieter Simoens *is a postdoctoral researcher in the Department of Information Technology at Ghent University, Belgium. His research focuses on the optimization of protocols to enable mobile cloud computing. Simoens received a PhD in electronic engineering from Ghent University. Contact him at pieter.simoens@intec.ugent.be.*

Filip De Turck *is a part-time professor and a postdoctoral fellow in the Department of Information Technology at Ghent University. His research interests include scalable software architectures for telecommunication networks and service management, and performance evaluation and design of new telecommunication services. De Turck received a PhD in electronic engineering from Ghent University. He is a member of IEEE and the ACM. Contact him at filip.deturck@intec.ugent.be.*

Bart Dhoedt *is a professor on the Faculty of Applied Sciences in the Department of Information Technology at Ghent University. His research interests include software engineering, distributed (autonomic) systems, grid and cloud computing, and thin-client computing. Dhoedt received a PhD in electronic engineering from Ghent University. He is a member of IEEE. Contact him at bart.dhoedt@intec.ugent.be.*

Piet Demeester *is a professor of communication networks at Ghent University, where he heads a research group active in the area of networks and distributed software and services (www.ibcn.intec.ugent.be). He received a PhD in electronic engineering from Ghent University. Demeester is an IEEE Fellow. Contact him at piet.demeester@intec.ugent.be.*



IEEE computer society

PURPOSE: The IEEE Computer Society is the world's largest association of computing professionals and is the leading provider of technical information in the field.
MEMBERSHIP: Members receive the monthly magazine *Computer*, discounts, and opportunities to serve (all activities are led by volunteer members). Membership is open to all IEEE members, affiliate society members, and others interested in the computer field.
COMPUTER SOCIETY WEBSITE: www.computer.org
OMBUDSMAN: To check membership status or report a change of address, call the IEEE Member Services toll-free number, +1 800 678 4333 (US) or +1 732 981 0060 (international). Direct all other Computer Society-related questions—magazine delivery or unresolved complaints—to help@computer.org.
CHAPTERS: Regular and student chapters worldwide provide the opportunity to interact with colleagues, hear technical experts, and serve the local professional community.
AVAILABLE INFORMATION: To obtain more information on any of the following, contact Customer Service at +1 714 821 8380 or +1 800 272 6657:

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

PUBLICATIONS AND ACTIVITIES
Computer: The flagship publication of the IEEE Computer Society, *Computer*, publishes peer-reviewed technical content that covers all aspects of computer science, computer engineering, technology, and applications.
Periodicals: The society publishes 13 magazines, 18 transactions, and one letters. Refer to membership application or request information as noted above.
Conference Proceedings & Books: Conference Publishing Services publishes more than 175 titles every year. CS Press publishes books in partnership with John Wiley & Sons.
Standards Working Groups: More than 150 groups produce IEEE standards used throughout the world.
Technical Committees: TCs provide professional interaction in more than 45 technical areas and directly influence computer engineering conferences and publications.
Conferences/Education: The society holds about 200 conferences each year and sponsors many educational activities, including computing science accreditation.
Certifications: The society offers two software developer credentials. For more information, visit www.computer.org/certification.

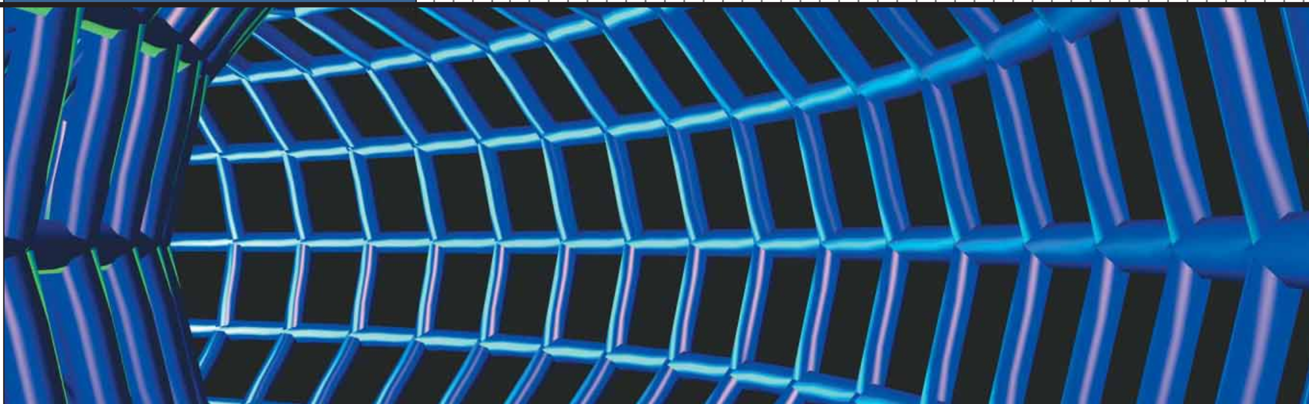
NEXT BOARD MEETING
13–14 Nov., New Brunswick, NJ, USA

IEEE

revised 18 July 2011

COMPUTER SOCIETY OFFICES
Washington, D.C.: 2001 L St., Ste. 700, Washington, D.C. 20036-4928
Phone: +1 202 371 0101 • **Fax:** +1 202 728 9614
Email: hq.ofc@computer.org
Los Alamitos: 10662 Los Vaqueros Circle, Los Alamitos, CA 90720-1314
Phone: +1 714 821 8380
Email: help@computer.org

MEMBERSHIP & PUBLICATION ORDERS
Phone: +1 800 272 6657 • **Fax:** +1 714 821 4641 • Email: help@computer.org
Asia/Pacific: Watanabe Building, 1-4-2 Minami-Aoyama, Minato-ku, Tokyo 107-0062, Japan
Phone: +81 3 3408 3118 • **Fax:** +81 3 3408 3553
Email: tokyo.ofc@computer.org



Using Mathematical Modeling in Provisioning a Heterogeneous Cloud Computing Environment

Sungkap Yeo and Hsien-Hsin S. Lee, *Georgia Institute of Technology*

Mathematical models demonstrate that to achieve optimal performance in a heterogeneous cloud infrastructure, the slowest node's response time should be no more than three times that of the fastest node.

Cloud computing is transforming the entire IT industry, high-performance computing (HPC), and personal data sharing and management. In cloud computing, computing power is supplied as a utility, similar to electricity or water. As such, service providers can centrally manage, maintain, and upgrade computing resources, offloading the burden from small business owners or those who do not have the expertise or budget to handle the fast-changing computing infrastructure.

Using the cloud for HPC can substantially reduce the total cost of ownership by eliminating the need to maintain large-scale parallel machines and their energy-consuming power and cooling systems.[1,2] From a cost-effectiveness perspective, there are tradeoffs in terms of resource provisioning given that a target task can be parallelized, a common case for throughput-oriented computing.

For example, assume that an HPC job, which can be perfectly parallelized, takes eight hours to complete using one computing node. If the cloud computing service provider charges for a job on a per-machine-hour basis (that is, based on the accumulated machine time), instead of running it on one node for eight hours, the job can be finished in one hour on eight machines with eight times speedup with the same utility charge (eight machine hours).

One trend that complicates this tradeoff is the heterogeneity in a cloud computing environment. Although a cloud service provider can start with near-homogeneous computing nodes, the facility will likely grow more heterogeneous over time due to upgrades and replacement. Therefore, not only will each computing node's performance and capability continue to deviate, the new computing nodes will also provide better performance for the same amount of power due to technology scaling and architectural innovation. Because of this heterogeneity, response times will vary significantly depending on provisioning policies. To mitigate this variation and guarantee quality of service, the cloud provider might want to dismiss the slowest computing nodes. The question is how slow a physical node can be for a given task to maintain its optimal computing quality in terms of execution time and energy cost.

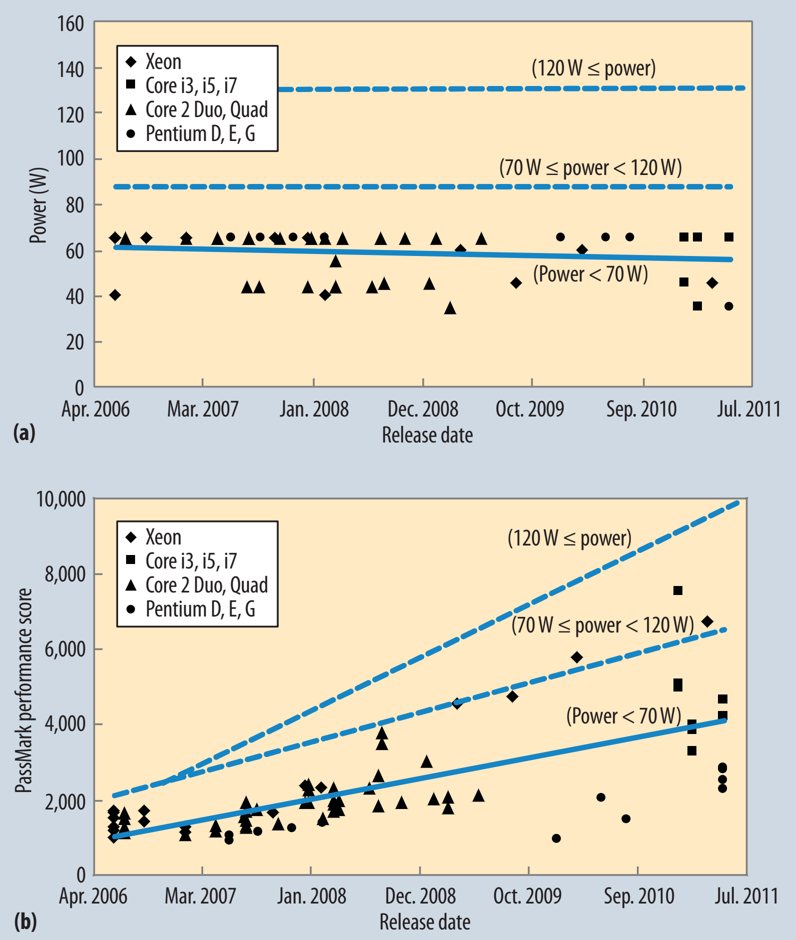


Figure 1. Power consumption and performance of Intel's CPUs since 2006: (a) power consumption and (b) performance. Over time, newer CPUs achieve higher performance than the older ones without compromising the power consumption.

To tackle this issue, we established a mathematical model based on statistics for a heterogeneous cloud environment. To understand optimal provisioning in a cloud, we used this model to evaluate the tradeoff of a task's execution time and energy consumption.

CLOUD COMPUTING MODEL

For this study, we assume the workload is perfectly parallelizable, which is often the case for throughput-oriented computing in HPC and transactional processing applications. For example, the most common cloud computing application is file transferring on the Web. Servers in the cloud can process all the requests received by a Web service at the same time individually and independently. Therefore, the cloud can achieve n times speedup when n nodes are deployed if and only if the number of concurrent users is always larger than or equal to n.

Next, we assume that an entire workload can be evenly divided into m smaller job units without affecting the workload's scalability. We also assume that m is larger than n, where n represents the maximum number of virtual machines in the cloud. (For simplicity, $m = kn$, where k is a positive integer.) In this study, one job unit represents the smallest task running to the end on a single physical node without interruption. However, we do not consider intermittent context switches within one job unit as interruption as long as the task keeps running on the same physical node.

In addition, we do not allow a virtual machine to be migrated among physical nodes during a job unit's execution because this migration will not only include the executable image but also all the architectural states, including the memory footprint. Data migration on interconnected cloud computing nodes would likely cause significant performance degradation due to peer-to-peer communication.

Cloud power and performance behavior

Before detailing power and performance in a heterogeneous cloud, we present a scenario from a cloud administrator's perspective. Typically, cloud service providers begin their cloud computing business with several near-homogeneous computing nodes. Over time, the cloud provider will replace some of the old computing nodes with newer nodes featuring the latest technologies. Gradually, the capability and performance of all machines in the cloud will become more heterogeneous. Although previous studies considered heterogeneity at the microarchitectural[3] and system levels,[4] they all assumed heterogeneity in the same generation of manufacturing technology. We consider computing heterogeneity in a broader sense.

We reviewed the power and performance trends of commercial microprocessors over the past few years and used our observations to justify our model assumption. We first plotted the thermal design power (TDP) numbers and the PassMark performance scores[5] for several processors under 70 W, including Pentium, Core 2, Core i3/5/7, and Xeon. This included all commercial desktop and server processors from Intel from January 2006 to February 2011, except Celeron processors and certain processors that did not report TDP or PassMark results. The solid line in Figure 1 shows their asymptotic power consumption and performance trend between 2006 and 2011. The dashed lines without individual dots show the trends of two other machine groups based on their TDP: 70 W to 120 W and more than 120 W.

We applied regression methods to estimate the relationship between power and performance over time. Taking all the samples into account, we plotted our regression models for power and performance (solid lines in Figure 1). As Figure 1b shows, the performance continues to improve for each machine group across different proliferations or generations. On the other hand, the TDP trend in Figure 1a shows negligible growth. More interestingly, the TDP trends for the two lower-power machine classes are decreasing. This decrease is the consequence of a recent awareness of the power wall, which gradually increases the heat dissipation cost. For the same reason, we anticipate that the power grade of future processors will remain below the bar. This also implies that with the same power budget, newer machines can deliver higher performance. In other words, performance per power (a metric derived by dividing the performance score by the power consumption) continues to grow over time. For example, the 95 W Core i7 (Lynnfield), released in September 2009, achieves higher performance than the 95-W Pentium D (Presler), released in January 2006. This difference is largely attributable to technological advances in microarchitecture as well as scaled-down feature size and supply voltage.

Given these observations, we define our model of power and performance for a future heterogeneous cloud based on two assumptions.

First, the computing nodes in the cloud we analyze are heterogeneous, having different microarchitectures fabricated using different processes. Thus, the cloud provides varied capability and process technologies. Second, the performance capabilities of these computing nodes are uniformly distributed (from low to high) but consume exactly the same amount of power.

The rationale behind this second assumption is twofold. First, for a given power budget, the performance of each machine class continues to improve linearly while their power envelope remains pretty much unchanged. In other words, the power efficiency measured by performance per power improves over time. Second, when a datacenter phases out some computing nodes due to an upgrade, it can safely deploy new computing nodes only when these upgrades' aggregated power consumption does not exceed the original. Otherwise, the datacenter must also upgrade its power delivery infrastructure as well as its cooling capacity to accommodate the new servers. Given this overhead, we anticipate that the replacement and upgrade will be done without altering the power delivery infrastructure. Therefore, we assume that the newly deployed servers will improve performance linearly across different machine proliferations while using the same amount of power.

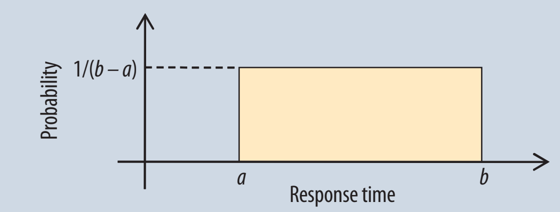


Figure 2. Probability distribution function (PDF) of the execution time of a job unit when there are *n* virtual machines. The execution time is uniformly distributed from *a* seconds (the fastest node) to *b* seconds (the slowest node).

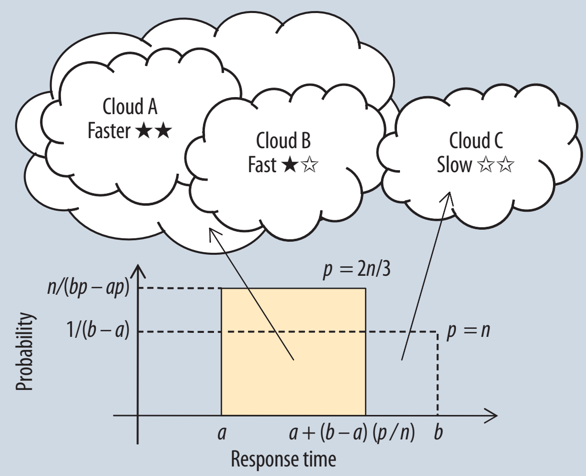


Figure 3. Probability distribution function (PDF) of the execution time of a job unit when there are 2*n*/3 virtual machines. The worst-case response time is improved by dismissing the one-third of physical nodes with the lowest performance.

To express this distribution mathematically, we assume that the response time for executing a job unit in such a cloud is uniformly distributed from a seconds (the fastest node) to b seconds (the slowest node). Figure 2 illustrates the probability distribution function (PDF) of the response time for executing a job unit in this cloud.

On the other hand, we assume that the cloud service provider can improve the worst-case response time by dismissing physical nodes with the least performance. For example, when a cloud service provider decides to retire one-third of its physical nodes from the slowest batch, we assume that the new response time for executing a job unit of this cloud becomes a uniform distribution from a seconds to $(a + 2b)/3$ seconds, represented by $U(a, (a + 2b)/3)$. As such, we assume that the maximum number of virtual machines that can be allocated on this cloud also shrinks in the same ratio.

Figure 3 shows the impact of retiring one-third of a cloud service provider's physical nodes from the cloud. The variable p in this figure represents the maximum number of virtual machines that can be allocated on the cloud, while

n represents the maximum number of virtual machines for the original cloud as shown in Figure 2. Moreover, the PDF in Figure 3 shows the improved worst-case response time as a result of removing one-third of the physical nodes from the slowest side.

Nevertheless, in the given response time PDF, we did not assume that a particular virtual machine can pick a physical node at a particular speed. Rather, when a cloud's PDF is given, we assumed that a virtual machine's behavior in this cloud follows the PDF in a statistical manner. In other words, we assumed that virtual machines will be uniformly distributed across the physical nodes.

Although dispatching more jobs to newly deployed servers with higher power efficiency increases energy efficiency, this is not the case for a datacenter, for two reasons. First, for a datacenter, it is important to balance the power draw across the AC phases.[6] The balance will break when jobs are distributed to only certain computing racks. Second, we want to minimize the number of hotspots for a datacenter, a common consequence of unbalanced workloads. Hotspots generally cause higher machine failure rates and require additional attention and effort to remove the heat.

Given that each computing node consumes the same amount of power, energy consumption will be proportional to the total execution time.

Execution time and energy consumption

We define the execution time of a given workload on a cloud as the time required to finish a workload consisting of m job units. When some job units are assigned to more than one virtual machine, the execution time, in our definition, is bounded by the virtual machine that finishes last. For example, when an animator renders a movie comprising m independent frames, the movie cannot be released before the last frame finishes rendering. In addition, when comparing the performance of cloud configurations, we use as the baseline the case of executing the same amount of workload on a virtual machine running on the fastest node. When we use more virtual machines to execute the workload in parallel, we use slower nodes to accomplish the task. As a result, the parallelized version could reduce the overall effectiveness of energy consumed in the cloud.

Energy consumption is the total energy needed to complete a given workload. In particular, when some physical nodes finish their assigned job units before the others, we assume that these nodes will not consume energy while waiting for the others to finish. This is because, in a real-world scenario, these nodes will either be assigned to other tasks or moved to a near-zero power state to save energy.[7] In addition, given that each computing node consumes the same amount of power, energy consumption as defined will be proportional to the total execution time. Therefore, we calculate a parallelized workload's utility consumption as the summation of each virtual machine's execution time.

To quantify the effectiveness of resource provisioning in a cloud, we use the energy-delay product (EDP),[8] which we calculate by multiplying the execution time (seconds) with the energy consumption (joules). We will use this metric in our subsequent evaluation when provisioning resources (that is, the number of virtual machines to allocate to achieve optimal energy efficiency).

ANALYTICAL EVALUATION

Next, we use analytical models, based on our assumptions, to compare each configuration's EDP to the baseline EDP.

Baseline

The baseline of our study assumes that the entire job is performed on one virtual machine running on the fastest physical node. In this case, the fastest physical node can retire a job unit every a seconds. Because there are m independent job units in the entire workload, the baseline configuration takes ma seconds to finish. This configuration consumes $W \times ma$ joules for completing the entire workload, where W represents a physical node's power. Thus, the EDP of this study's baseline is

$$EDP_{base} = (W \times ma)(ma) = Wm^2a^2.$$

Expectation-based analysis

We use an expectation-based analysis to determine a cloud model's execution time and energy consumption. We use a new distribution function to represent the execution time of a virtual machine with more than one job unit.

Execution time distribution across virtual machines. The PDF of the response time when using p virtual machines is given by $U(a, (a + ((b - a)p)/n))$, as Figure 3 illustrates. However, when a virtual machine is responsible for more than one job unit (that is, m/p units), the virtual machine's total execution time cannot be modeled the same way. Rather, we model it as the summation of independently selected m/p samples from Figure 3. When we add independent samples from a uniform distribution, the summation's distribution function tends to approach a normal distribution according to the central limit theorem.[9] This theorem proves that when we add more independent samples into the summation, the summation's distribution will become more like a normal distribution. In addition, the summation of 12 samples is known to be good enough

Computer | Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page

to satisfy the central limit theorem.[9] In this case, we assume that a virtual machine is responsible for more than 12 job units by letting $m \geq 12n$ (that is, $m \geq 12p$ because $p \leq n$).

Now our goal is to obtain the mean and variance of the normal distribution representing the total execution time of a virtual machine responsible for m/p job units. First, we calculate the mean and variance for the original uniform distribution, $U(a,(a + ((b - a)p)/n))$:

$$\text{Mean} = \frac{1}{2}\left(a + a + \frac{(b-a)p}{n}\right) = a + \frac{(b-a)p}{2n} \text{ and}$$

$$\text{Variance} = \frac{1}{12}\left(a + \frac{(b-a)p}{n} - a\right)^2 = \left(\frac{(b-a)p}{\sqrt{12}n}\right)^2$$

The central limit theorem shows that the summation of m/p independent samples from this distribution will become a normal distribution with the following mean and variance:

$$N\left(\frac{m}{p}\left(a + \frac{(b-a)p}{2n}\right), \left(\sqrt{\frac{m}{p}} \times \frac{(b-a)p}{\sqrt{12}n}\right)^2\right) = N(\mu,\sigma^2) \quad (1)$$

For convenience, we use μ and σ^2 to denote the distribution's mean and variance. All in all, when using p virtual machines, each machine's execution time will follow the normal distribution, $N(\mu, \sigma^2)$. The ultimate question is, "How many seconds will it take to finish the entire workload?" To answer this question, we must first determine the expectation of the largest sample from $N(\mu, \sigma^2)$ when we must pick p samples. Because the overall execution time depends on the slowest virtual machine that finishes last, the largest of p samples will give the total execution time.

Expectation of the largest sample. Before finding the largest sample's expectation, we discuss the same question for the standard normal distribution, $N(0, 1)$. Let $pdf(x)$ be the PDF of the standard normal distribution. In this PDF, let y be the largest sample among randomly chosen p samples. For each case out of p cases, the probability of y being the largest sample is given as follows:

$$\text{Probability} = pdf(y) \times \left(\int_{-\infty}^{y} pdf(x)dx\right)^{p-1}$$

Equation 2 gives the expectation of the variable y.

$$\int_{-\infty}^{\infty} p \times y \times pdf(y) \times \left(\int_{-\infty}^{y} pdf(x)dx\right)^{p-1} dy = ExB(p) \quad (2)$$

For convenience, $ExB(p)$ denotes the expectation of the largest sample among p samples from the standard normal distribution. In addition, by substituting $pdf(x)$ in Equation 2 with Equation 3, we can find the numerical values of $ExB(p)$ for various p. We show the results in the middle column of Table 1.

Table 1. Expectation of the largest sample (*ExB*(*p*)) from *N*(0, 1).

Number of samples (*p*)	Value using Equation 2	Value using the experimental method
1	0.00000	–0.00001
2	0.56419	0.56419
4	1.02938	1.02938
8	1.42360	1.42356
16	1.76599	1.76591
32	2.06967	2.06968
64	2.34373	2.34368
128		2.59461
256		2.82679
512		3.04392

$$pdf(x) = \frac{1}{\sqrt{2\pi}}\exp\left(-x^2/2\right) \quad (3)$$

Because Equation 2's complexity grows exponentially as p increases, we cannot find the exact numerical values of $ExB(p)$ for $p > 64$. To address this shortcoming, we propose a more scalable way of approximating the values in Table 1. In this solution, we first implement a random number generator that produces random numbers from the standard normal distribution. Using this random number generator, our solution picks p independent random samples and remembers the largest sample among them. This operation continues for a long enough time (for example, to produce the results in Table 1, our software repeated this operation more than 100 million times) and averages the largest samples. This experimental method generates the exact numerical values of $ExB(p)$, as shown in the third column of Table 1, after averaging more than 100 million trials. As a comparison of the second and third columns in the table shows, the mathematical accuracy is slightly compromised in exchange for scalability. However, we do not expect the tiny error rate to affect our analysis and conclusion.

The study of the largest sample in the standard normal distribution gives us an idea about the $ExB(p)$ for other normal distributions. Let a random variable X follow $N(\mu, \sigma^2)$ with $\mu \neq 0$, $\sigma \neq 1$, $\sigma \neq 0$, and a derived random variable $Y = (X - \mu)/\sigma$. Then, Y follows $N(0, 1)$ by recalling the property that if X follows $N(\mu, \sigma^2)$ and a and b are real numbers, then $aX + b$ follows $N(a\mu + b,(a\sigma)^2)$. From Equation 2, the expectation of the largest sample for Y is $ExB(p)$ because $Y = (X - \mu)/\sigma, X = Y\sigma + \mu$; and the expectation of the largest sample for X is $ExB(p) \cdot \sigma + \mu$. Now, we can calculate the expectation of the largest sample for any arbitrary normal distribution.

Execution time and energy consumption analysis. In our model, each of the p virtual machines is responsible for m/p job units, and the response time for each job unit follows $U(a, (a + ((b - a)p)/n))$. We use the following equation to calculate the expectation of the time required on a virtual machine finishing last:

$$\begin{aligned}&\text{Execution time}\\&= \mu + ExB(p) \times \sigma\\&= \frac{m}{p}\left(a + \frac{(b-a)p}{2n} + ExB(p) \times \sqrt{\frac{m}{p}} \times \frac{(b-a)p}{\sqrt{12}n}\right)\\&= \frac{ma}{2np}\left(2n + \left(\frac{b}{a}-1\right)p + ExB(p) \times p^{3/2}\sqrt{\frac{1}{3m}} \times \left(\frac{b}{a}-1\right)\right)\\&= \frac{ma}{2np}\left(2n + \left(\frac{b}{a}-1\right)p + Unbalance\left(\frac{b}{a}, p, m\right)\right)\end{aligned}$$

In this equation, we name the second term *unbalance*, which becomes zero if and only if every virtual machine finishes at the same time:

$$Unbalance\left(\frac{b}{a}, p, m\right) = ExB(p) \times p^{3/2}\sqrt{\frac{1}{3m}} \times \left(\frac{b}{a}-1\right) \quad (4)$$

For example, a higher deviation from the normal distribution indicates that the random samples from this distribution are more spread out, increasing the probability of having more deviated samples. In our case, because we model a virtual machine's finishing time by picking a sample from Equation 1, more deviated samples indicate that the workload assignment is unbalanced among virtual machines executing this workload. In particular, a larger b/a will lead to a larger σ^2 in Equation 1 and a larger $Unbalance((b/a), p, m)$ in Equation 4. Hence, we can conclude that a larger b/a value causes a more unbalanced workload distribution among virtual machines, degrading the overall performance. Also note that $Unbalance((b/a), p, m)$ is directly proportional to $1/\sqrt{m}$. Because m is independent of p and b/a, changing the value of m will not affect other variables in Equation 4. This implies that a very large m will eventually zero out Equation 4. Thus, we can express the execution time when $m \to \infty$ as

$$\begin{aligned}&\text{Execution time } (when\ m \to \infty)\\&= \frac{ma}{2np}\left(2n + \left(\frac{b}{a}-1\right)p\right)\end{aligned}$$

We also evaluate the energy consumption probabilistically. Because performance is bounded by the execution time of the virtual machine finishing last, we must calculate the expectation of the largest sample from Equation 1. In contrast, to evaluate the utility consumption, we must focus on the average execution time of p virtual machines. This is because, in a normal distribution, the probability for having $\mu + \alpha$ samples is exactly the same as having $\mu - \alpha$ samples. This fact indicates that the odds of having a virtual machine consuming α seconds more than the average is the same as having a virtual machine consuming α seconds less than the average. Therefore, we conclude that the expectation of the total execution time is given by $\mu \times p$, the number of virtual machines. Given the power of a physical node in the cloud is W, the total energy consumption will be as follows:

$$\begin{aligned}\text{Energy consumption} &= W \times \frac{m}{p}\left(a + \frac{(b-a)p}{2n}\right)p\\&= W \times m\left(a + \frac{(b-a)p}{2n}\right),\end{aligned}$$

$$\begin{aligned}&EDP_{\text{exp}}(p) = \frac{Wm^2a^2}{4n^2p} \times\\&\left(\left(2n + \left(\frac{b}{a}-1\right)p + Unbalance\left(\frac{b}{a}, p, m\right)\right)\left(2n + \left(\frac{b}{a}-1\right)p\right)\right)\\&= \frac{EDP_{\text{base}}}{4n^2p} \times\\&\left(\left(2n + \left(\frac{b}{a}-1\right)p + Unbalance\left(\frac{b}{a}, p, m\right)\right)\left(2n + \left(\frac{b}{a}-1\right)p\right)\right)\end{aligned}$$

Similarly, we calculate the EDP for $m \to \infty$ as follows:

$$EDP_{\text{exp},m \to \infty}(p) = \frac{EDP_{\text{base}}}{4n^2p} \times \left(2n + \left(\frac{b}{a}-1\right)p\right)^2 \quad (5)$$

To visualize the effect of a large m in the EDP_{exp} metric, Figure 4 shows the EDP analysis for $m = 12n$, $m = 120n$, and $m \to \infty$ using the following coefficients: $n = 16{,}384$, $b/a = 1, 2, 3, 5$, and $ExB(p)$ from Table 1. To find the exact value p that makes the EDP metric a global minimum point, we take the derivative of Equation 5 with respect to p and set it to zero:

$$\frac{d}{dp}\left(EDP_{\text{base}} \times \frac{(2n + (b/a - 1)p)^2}{4n^2p}\right) = 0$$

$$p = \frac{2n}{\frac{b}{a} - 1}\ (\because p > 0) \quad (6)$$

In the example of $m \to \infty$ in Figure 4, we achieve the minimum EDP when $p = 2n/(b/a - 1) = 16{,}384$ in Figure 4c or $p = 2n/(b/a - 1) = 8{,}192$ in Figure 4d.

Again, $p = n$ must be fulfilled while maintaining Equation 6 to be energy-effective for all n virtual machines in the cloud. By combining two conditions, $p = n$ and Equation 6, we can calculate the requirement of b/a as $n = 2n/(b/a - 1)$; $b/a = 3$. This equation suggests that in a heterogeneous cloud computing environment with uniformly distributed performance, physical nodes that respond 3 times slower than the fastest node should not be used when attempting to minimize the EDP.

These models and analysis can be used to guide future deployment, allocation, and upgrades of cloud infrastructure to achieve optimal utility effectiveness. The findings presented here hold not only for a computing

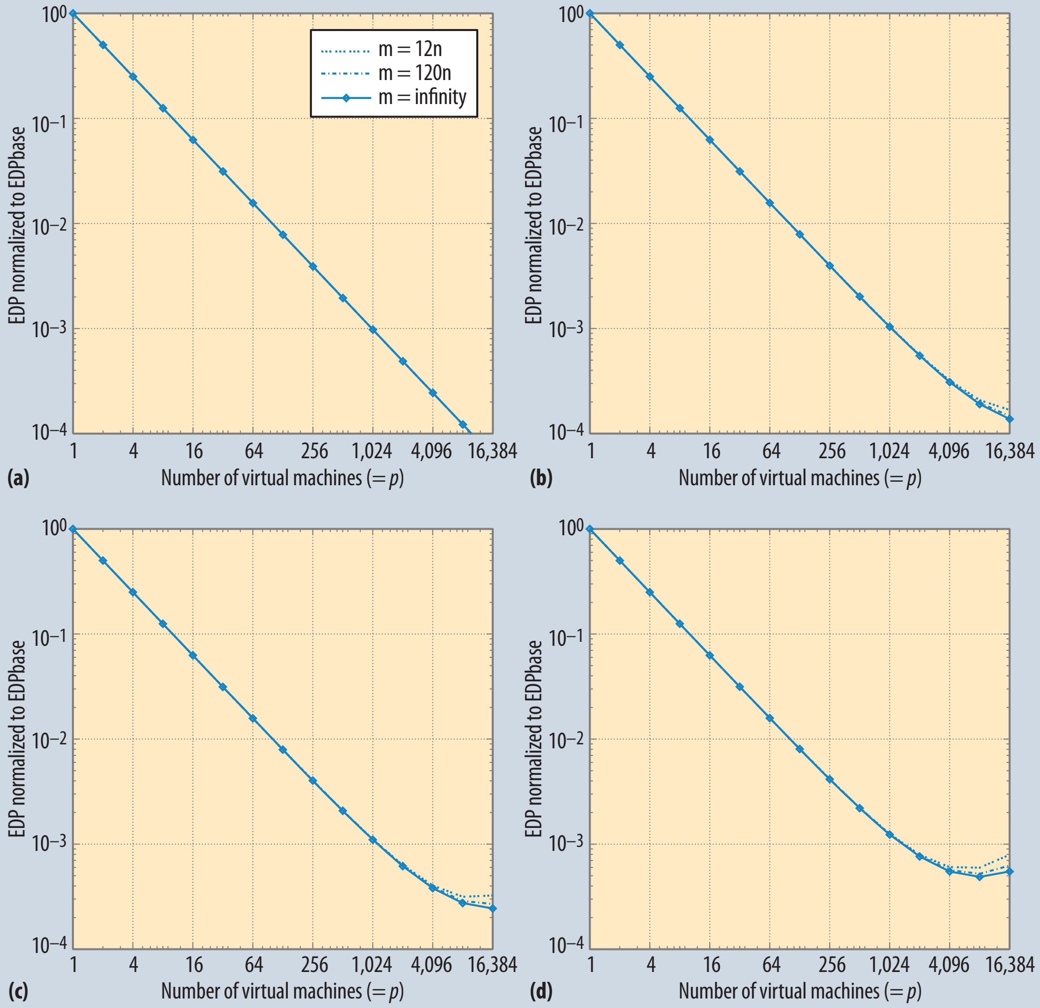


Figure 4. Example of the expectation-based analysis where the total number of available virtual machines is 16,384: (a) $b/a = 1$, (b) $b/a = 2$, (c) $b/a = 3$, (d) $b/a = 5$. When the response time of the slowest node b divided by that of the fastest node a is larger than 3, using all available virtual machines will compromise EDP.

environment operated by a single heterogeneous datacenter but also for a larger computing service comprising many datacenters of varying ages. Because more recent datacenters show better energy efficiency, the effectiveness of the collaboration of multigenerational datacenters can be analyzed in the same way. Our future work will study the power management techniques and scheduling algorithms for the heterogeneous cloud computing environments. C

References

1. M. Palankar et al., "Amazon S3 for Science Grids: A Viable Solution?" *Proc. 2008 Int'l Workshop Data-Aware Distributed Computing*, ACM Press, 2008, pp. 55-64.
2. L.A. Barroso, "The Price of Performance," *ACM Queue*, vol. 3, no. 7, 2005, pp. 48-53.
3. S. Ghiasi, T. Keller, and F. Rawson, "Scheduling for Heterogeneous Processors in Server Systems," *Proc. 2nd Conf. Computing Frontiers*, ACM Press, 2005, pp. 199-210.
4. R. Nathuji, C. Isci, and E. Gorbatov, "Exploiting Platform Heterogeneity for Power-Efficient Datacenters," *Proc. 4th Int'l Conf. Autonomic Computing* (ICAC 07), IEEE CS Press, 2007, pp. 5-14.
5. PassMark Software, "CPU Benchmarks;" www.cpubenchmark.net.
6. S. Pelley et al., "Power Routing: Dynamic Power Provisioning in the Datacenter," *Proc. 15th Int'l Conf. Architectural Support for Programming Languages and Operating Systems* (ASPLOS 10), ACM Press, 2010, pp. 231-242.
7. D. Meisner, B.T. Gold, and T.F. Wenisch, "PowerNap: Eliminating Server Idle Power," *Proc. 14th Int'l Conf. Architectural Support for Programming Languages and Operating Systems*, (ASPLOS 09), ACM Press, 2009, pp. 205-216.

8. R. Gonzalez and M. Horowitz, "Energy Dissipation in General Purpose Processors," *IEEE J. Solid-State Circuits*, vol. 31, no. 9, 1996, pp. 1277-1284.
9. J.A. Rice, *Mathematical Statistics and Data Analysis*, Duxbury Press, 2007.

***Sungkap Yeo** is a PhD student in electrical and computer engineering at the Georgia Institute of Technology. His research interests include power management for datacenters and microarchitectural support for the future cloud computing environment. Yeo received an MS in electrical and computer engineering from Georgia Institute of Technology. Contact him at sungkap@gatech.edu.*

***Hsien-Hsin S. Lee** is an associate professor in the School of Electrical and Computer Engineering at the Georgia Institute of Technology. His research interests include computer architecture, cybersecurity, and 3D integration. Lee received a PhD in computer science and engineering from the University of Michigan at Ann Arbor. He is a senior member of IEEE and the ACM. Contact him at leehs@gatech.edu.*

From the Presidents of the IEEE Computer Society and ACM

Alain Chesnais, *President, ACM*
Sorel Reisman, *President, IEEE Computer Society*

The overlapping interests of the IEEE CS and the ACM include their support of lifelong computing education programs.

As most members of our organizations probably know, there is a great deal of overlap between the products and services offered by the IEEE CS and ACM. In fact, about 24 percent of IEEE CS members belong to ACM, and 27 percent of ACM members belong to IEEE CS.

This became very clear to us when we first met at the SC2010 Conference in New Orleans, and had a brief conversation before stepping onstage to present the conference awards. Clearly, this wasn't a serendipitous meeting, as the SC Conference is sponsored annually by both organizations. What did surprise us was to learn that one of us (Alain), while born in Paris resides in Toronto, and the other (Sorel), resides in California but was born in Toronto. Moreover, we both have French names!

Two months later, over lunch in Toronto, we began to explore ways we might capitalize on the healthy competition our organizations have shared that has served the profession well over the decades. In a subsequent breakfast meeting three months later, we came up with the idea to craft a joint message to our members to appear concurrently in the August issue of *Communications* and *Computer*. Our goal is twofold: to describe areas in which we have cooperative and synergistic activities, and to explore new areas of mutual interest— all for the betterment of our profession.

One of the interesting challenges in writing this letter together has been how to present it without one or the other appearing to be "in control." Rather than flipping a coin to determine whose name should come first, we decided we would alternate leads—with Sorel as the lead author in the *Communications*-based version and Alain taking the lead here in *Computer*.

As mentioned earlier, there are many areas in which the two organizations have overlapping interests, but the one for which both associations are internationally recognized is computing education. While our strategic strengths in this area may appear to have different foci—with ACM's being academic education geared toward schools and universities and the IEEE CS's efforts centered on professional education—in fact both organizations support lifelong computing education programs. For example, both ACM and IEEE CS

- have representatives in the Computing Research Association (CRA), a premier resource for information about computer science education and technology-related public policy in the US;
- have representatives on the International Federation of Information Processing (IFIP);
- share metadata in respective digital libraries;
- promote Computer Science Education Week (Dec. 5-11, 2011);
- developed and maintain the widely adopted curricula for computing, computer engineering, computer science, software engineering, and information science;
- sponsor the graduate software engineering curriculum;
- are working together to develop the Partnership for Advancing Computing Education (PACE), to be recognized as a cross-organization unified voice in support of computing education; and
- sponsor the aforementioned annual SC Conference (Nov. 12-18, 2011, Seattle, WA), the International Conference on Software Engineering (June 2-9, 2012, Zurich, Switzerland), along with many other conferences.

This letter to our joint membership is the first of what we hope will be many more cooperative activities aimed at capitalizing on the strengths of our organizations. We intend to continue to discuss new ideas about other joint efforts. More importantly, we would really like to hear from you, our members, with ideas and suggestions you might have that will help us along this path. To facilitate exchanges of ideas regarding this message, we set up electronic forums at http://cooperation.computer.org and http://cooperation.acm.org. Please visit those sites and participate in the discussions there.

In today's environment, we recognize that resources are constrained, and are likely to be so for a very long time. We also recognize the technologies we have invented and continue to invent are changing the world at a pace unlike any point in history. If ever there was a time for us to cooperate, it is now. ■

Alain Chesnais *is president of ACM and founder of Visual Transitions, Toronto, Canada. Contact him at chesnais@acm.org.*

Sorel Reisman *is president of IEEE CS and a professor at California State University, Fullerton. Contact him at sreisman@computer.org.*

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

IEEE computer society

IEEE Computer Society Election

Nominees for Computer Society Officers and Board of Governors Positions in 2012

On the following pages are the position statements and biographies of the IEEE Computer Society's candidates for president-elect, first and second vice presidents, and Board of Governors. Within each category, candidates are listed in alphabetical order. Election of officers to one-year terms and of Board members to three-year terms, each beginning 1 January 2012, will be by vote of the membership as specified in the bylaws. Ballots must be returned no later than 12:00 noon EDT on Tuesday, **4 October**.

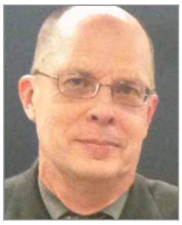
This year, the Computer Society will not mail paper ballot packages to all eligible voting members. Only members without an e-mail address in their member record, or who have opted out of IEEE e-mail communications, will receive a paper ballot package. Members who elect to use paper ballots should return them by mail to the IEEE Computer Society, c/o Survey & Ballot Systems, PO Box 46430, Eden Prairie, MN 55344-9876, USA, or by fax to Survey & Ballot Systems at +1-952-974-5110.

All other members will receive a broadcast e-mail message with their Web ballot package information. Members in all regions can vote via the Web at www.computer.org/election2011. For replacement ballots or to request a paper ballot, call +1-202-371-0101.

The opinions expressed in the statements are those of the individual candidates and do not necessarily reflect Computer Society positions or policies.

Results will be announced in the December issue of *Computer*.

NOMINEES FOR PRESIDENT-ELECT



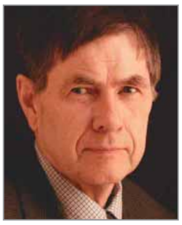
David Alan Grier

Position statement. Three words: Operations. Alliances. Communications. If you want a picture of the skills and ideas that I would bring to the Society presidency, you will find the most detail at my webpage, http://candidate.dagrier.net, or in my column, The Known World, in *Computer*. However, if you want the summary of my approach, you will find it in those three ideas: efficient operations, strategic alliances, and effective communications.

To the operations of the Society, I would bring skills honed for 20 years as a university administrator. The Society operates as a volunteer organization that needs to support the professional practice of 85,000 individuals with diverse goals and aspirations. A successful president needs to honor the contributions of the 85,000 members and their volunteer contributions to the society. The issues that the next president needs to consider include conference support, intellectual property rights, technical education, and plans for the Society's future. As a vice president, I drafted the plans for special technical committees and have been active on the planning and financial committees.

Alliances are crucial because no organization thrives without cooperating with institutions that have similar

(Grier continued on next page)



Jon Rokne

Position statement. The Computer Society is facing opportunities and challenges in several areas.

- *Membership*. Developing new membership benefits is a high priority in view of declining membership. This might include further incentives to attract new student members. Retaining student members as full members is also an issue that needs attention.
- *Accreditation*. The Computer Society has developed excellent accreditation tools and services for the computing community. I will work toward acceptance and recognition of these accreditation efforts and enlist the help of IEEE-USA in gaining government support for accreditation within the US.
- *Literacy*. The Computer Society should seize the opportunity to provide educational material that supports the development of increased computer literacy.
- *Conferences*. Conferences provide the main forum for in-person member contacts and the exchange of technical information. They are a vital component of Computer Society activities. Maintaining the viability of the conference program is therefore a priority.
- *Internationalization*. I consider it a priority to engage

(Rokne continued on next page)

0018-9162/11/$26.00

Computer Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page Qmags THE WORLD'S NEWSSTAND®

(Grier continued from page 65)

goals and aspirations. Our most important alliance is with the IEEE, and we need to build on the relationships that we have established in recent years. However, we would also benefit by strengthening our ties with other organizations in the fields of computer science, information technology, and education.

Finally, to the presidency I would bring the skills of a practiced communicator and one who understands the vital role of the Society. I would be able to help the Society better engage its members, other computer professionals, entrepreneurs, other members of the global economy, and also the general public. My success in communicating the value of an organization is seen in the writing that I have done for Computer Society publications.

Three ideas: strong organizational leadership, careful management of strategic alliances, clear and focused communication. This is what I would bring to the Computer Society as president. You can find more at http://candidate.dagrier.net.

Biography. David Grier's career in computer science began when he learned to program the Univac 1103 that was in his father's office. This career has encompassed many roles in the field: industry programmer, professor and researcher, entrepreneur, conference organizer, university administrator, consultant, and author. Grier's book, *When Computers Were Human* (Princeton), received a 2006 award for the best book on computing from the Independent Press Association.

Grier's computer education began with training courses at Univac and Burroughs. He passed through Knuth's *Art of Computer Programming* and received a PhD in mathematical statistics from the University of Washington. At George Washington University, Grier has taught computer science, served as assistant dean of engineering and associate dean of international affairs, and is currently an associate professor of international science and technology policy.

Grier is first vice president of the Computer Society and serves on its Board of Governors (2009-), Executive Committee (2009-), and Planning Committee (2010-). He is currently vice president of publications and has served on such IEEE committees as TAB Periodicals, Periodical Review, and the Publication Services and Products Board. Grier is a senior member of IEEE. A more detailed biography can be found at http://candidate.dagrier.net.

(Rokne continued from page 65)

the Computer Society's international membership more effectively in Society activities. One way of doing this would be to create online communities to discuss issues of interest to all Computer Society members.

- *Publications.* One of the main incentives for joining the Computer Society has been its publishing program. Members typically can subscribe to IEEE and Society publications at a significantly reduced cost. The CSDL and IEL electronic libraries have, however, made individual subscription incentives of less value to many members, due to institutional subscriptions. I would therefore encourage the development of new publishing initiatives that lead to member retention.
- *Open access.* Open access has clear benefits for Computer Society members and the community in general, since it results in more freely available information. The challenge is to make open access economically viable.
- *IEEE.* I would encourage cooperation within the Technical Activities Board and with other major IEEE boards. I also advocate that the Computer Society avail itself of services offered by IEEE when they are advantageous to the Society.

An expanded discussion of these topics is found at www3.telus.net/public/jrokne/public.html.

Biography. Jon Rokne is the second vice president and secretary of the IEEE Computer Society Board of Governors, past vice president of the IEEE Publication Services and Products Board, chair of the PSPB's Nominations and Appointments Committee, and a member of the IEEE Ethics and Members Conduct Committee.

Rokne has completed two terms as vice president of publications for IEEE and two terms as vice president of publications for the IEEE Computer Society. He has also served as a member of PSPB, PSPB Financial Committee, and PSPB Operations Committee, and as chair of a PSPB subcommittee on publications conduct.

A Computer Society Golden Core member, Rokne has served as a member of the Publications Board, chair of the Transactions Operations Committee, and chair of an ad hoc committee for ReadyNotes.

Rokne is a professor and former chair of the computer science department at the University of Calgary. He has published extensively in mathematics, including three jointly authored books. Rokne has also published in the areas of physically- and biologically-based computer simulations of leaves, auroras, ball lightning, social networks, and one jointly authored book, *Light Interaction with Plants* (Woodhead, 2004). In 2003, he organized the Pacific Graphics conference.

For further information, visit www3.telus.net/public/jrokne/public.

NOMINEES FOR FIRST VICE PRESIDENT

Thomas M. Conte

Position statement. This is one of the most challenging times in our profession. The global recession and its impact on our industry and on our lives have been profound. The IEEE Computer Society is here to support the computer professionals who make up the Society. This is important now more than ever. We can't let red tape or petty squabbles get in the way of a Computer Society that's useful and valuable to its members. I've been an active member of the Computer Society for more than two decades, serving in many leadership roles.

The Computer Society's lifeblood is our conferences, symposia, and technical committees. I believe strongly that these activities must have a higher profile within the Computer Society. Any legislative or administrative walls that have built up over the years should be torn down. Our members who are trying to run a conference, build a new technical committee's community, or expand an already vibrant community need our direct support. They should not hear "you can't do it," but instead hear "how can we help you do it?" The "Is it helping our members?" test is the guiding principle behind any and all decisions I have made as a Board of Governors member, and will be the same principle I use as vice president of the Society. This is our Society. I volunteer to the Computer Society to make sure it remains pointed in the right direction, and stays our Society.

Biography. Thomas M. Conte has been deeply involved with the Computer Society throughout more than 25 years of IEEE membership. He has shepherded major symposia as chair for the Technical Committee on Microarchitecture (TC-uARCH) and has served on editorial boards for *Computer* and *Micro* magazines, and *Transactions on Computers*, among others. Conte, an elected member of the Board of Governors, has served as the IEEE Computer Society Awards Committee chair since 2009. He has focused on streamlining and revamping the awards process so that more of our outstanding members are recognized for their accomplishments.

Conte is an IEEE Fellow and serves as chair of the 2011 IEEE Computer Society Fellows Evaluation Committee. He is a professor at the Georgia Institute of Technology, where he holds a joint appointment in the schools of computer science and electrical and computer engineering. Conte likes to say that, if given a choice between the intrigue of academic power brokering or the challenge of teaching sophomores K-maps and pipelining, he'd choose the latter hands-down, every time. His research focuses on many-core computer architecture, compiler code generation, and fast simulation techniques. Conte received a PhD in electrical engineering from the University of Illinois, Urbana-Champaign.

Jean-Luc Gaudiot

Position statement. I strongly believe in the Computer Society's mission as an international organization to provide technical information and services to advance the theory, practice, and application of information processing science and technology. If elected first vice president, I will continue to promote the services that are our hallmark, including transactions, magazines, and conferences.

We know how our profession has revolutionized the way in which scientific and engineering knowledge is created, disseminated, and evaluated. Most of our publications are now online and many are even available on a variety of e-readers. As the technology that our work has helped to create continues reaching researchers and practitioners in more disciplines in more new ways, several questions become more important to deal with: On what devices will we be receiving our articles? What new areas must we own to remain at the forefront of knowledge creation? Admittedly, these are but the tip of the iceberg.

I know what challenges await, and I have supported some of these efforts in several offices within the Computer Society. I submit that I am prepared to be first vice president and intend to serve as a unifying force to help coalesce opportunities into a coherent vision of our future. With all the challenges in so many dimensions in front of us, I will help the Computer Society continue to make a difference in the lives of its members. I ask you to give me a chance to help us do that.

Biography. Jean-Luc Gaudiot, currently a professor at the University of California, Irvine, has served the Computer Society for almost 20 years in a variety of offices, which has given him experience in many activities of the Society. He currently serves on the Board of Governors and the Audit Committee, and is chair of the Publications Board's Transactions Operations Committee. He was one of the founders and the first editor in chief of *IEEE Computer Architecture Letters*, editor in chief of *Transactions on Computers*, chair of the IEEE Computer Society Technical Committee on Computer Architecture for two terms, and program committee and general chair of many major conferences. Gaudiot is a Computer Society Golden Core member and a Fellow of IEEE and the AAAS.

Before joining UCI, where he was department chair for six years, Gaudiot was a professor at the University of Southern California. His industrial experience includes software and architecture design at Teledyne and TRW. Gaudiot's research focuses on computer architecture, a field in which he has more than 200 refereed publications.

In his spare time, he combines his passion for aviation with his love for teaching by working as an active flight instructor (both primary and instrument).

Computer Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page Qmags THE WORLD'S NEWSSTAND®

NOMINEES FOR SECOND VICE PRESIDENT

Paul Croll

Position statement. In the more than 25 years that I have been a Computer Society volunteer, I have demonstrated an ability to understand Society-related markets, engage key leaders from communities of interest, and establish collaborative alliances with other IEEE and external organizations. If elected, I will bring that same forward-looking, business-based, service-oriented, strategic focus to the Executive Committee to help sustain our reputation for excellence and support our growth in a climate that's fast-changing in terms of technology and economics.

The continued growth and success of the Computer Society will depend upon how effectively we leverage our core technical excellence for the benefit of a broader constituency than we might have imagined in the past. Our growth is also dependent upon how well we nurture and grow our chapters, providing a strong local presence and engaging both students and practitioners throughout the globe. To thrive, we must embrace new mechanisms for reaching and serving our communities of interest, while maintaining our core foundation as the world's most prestigious professional society in the field of computing. This means that while we establish and nurture new Web-based communities and new electronic products and services designed to reach and engage a broader constituency, we must also take care to nurture and grow our technical core of conferences and publications.

We are the world's leading organization for computing professionals. I will work hard to ensure that we continue to serve our community with offerings that are relevant, provide technical value, are accessible, and are affordable.

Biography. A Fellow and chief scientist at CSC, Paul Croll has more than 35 years' experience in software and systems engineering. His experience spans industry, government, and academia as a practitioner, researcher, and university lecturer. Croll has been a Society volunteer for more than 25 years and is an active leader in establishing the Society's technical agenda.

Under his leadership, the Software and Systems Engineering Standards Committee maintains the largest collection of software and systems engineering standards in the world, as well as strategic liaisons with ASQ, INCOSE, PMI, and SEI. As the current vice president for technical and conference activities, Croll oversees the activities of more than 40 technical committees and 200 conferences. He leads efforts in the broader technical community as chair of the National Defense Industrial Association's software industry experts panel and as industry cochair for the DHS/DoD/NIST Software Assurance Forum's working group on processes and practices. Previously, he served as chair of the Technical Council on Software Engineering.

A Computer Society Golden Core member and Distinguished Visitor, Croll brings to the Society a demonstrated ability to bring together researchers, educators, students, and practitioners to advance both the state of the practice and the goals of the Society.

André Ivanov

Position statement. The Society needs to focus on serving its members and volunteers. Members, existing or prospective, need to be able to identify distinct value propositions made available to them by joining and remaining as active members. Also, members ought to see value in the Society's staff contributions. There should be opportunities to improve the alignment between membership needs and expectations and the Society's ways of conducting its businesses globally. To many non-North American members, the Society is too USA-centric. This should change; otherwise the Society will fail to reach its vision. Some current Society practices seem to deter enthusiasm and buy-in from volunteers. We need to acknowledge differences that exist between industrial practitioners and academics around the world.

The Society needs to ensure that its revenues remain uniquely directed to better serve the membership. The Society's efficiencies need improvement. Equally important, the Society needs to identify revenue models that are forward-looking yet focused on serving members. The Society should resist embarking down paths of complex business propositions that offer value in very risky and indirect fashion to members, again, especially those who reside outside North America. We need to improve communication with our members and engage with our volunteers such that they clearly see and leverage the benefits that the Society brings to their efforts and aspirations. If elected, I will work at making the Society shift its perspective to be even more global than it is today, focusing on offering clearly identifiable value to its members.

Biography. André Ivanov is head of electrical and computer engineering at the University of British Columbia. He has worked at PMC-Sierra and the University of Montpellier, the University of Bordeaux, and Edith Cowan University. Ivanov's research interests are in VLSI and systems-on-a-chip. He has published more than 200 papers and holds four US patents. In 2001, Ivanov cofounded semiconductor IP company Vector 12, and has served since 2006 as associate editor for *IEEE Transactions on Computer-Aided Design* and since 2004 for *Design & Test*.

Ivanov served on the Computer Society Board of Governors in 2008-2010, and is now serving a second three-year term. He served as vice chair of the Computer Society Fellows Committee in 2007 and as its chair in 2008. Ivanov served as program chair (2002) and general chair (2003, 2004) of the VLSI Test Symposium. He has long served the Test Technology Technical Council in leading roles, including a pair of two-year terms as elected chair (2004-2007).

Ivanov's other Computer Society roles include:

Conferences and Tutorials Board: vice-chair, 2005; member at large, 2007

Conference and Technical Activities Board: member at large, 2010, 2011

Technical Activities Board: Executive Committee member, 2004-2007

Audit Committee: member, 2011

BOARD OF GOVERNORS NOMINEES (12 NOMINEES; VOTE FOR SEVEN)

José-Ignacio Castillo-Velázquez

Position statement. I wish to continue being a Computer Society Board of Governors member to contribute my experience and passion to building the future of the Society and spreading word of IEEE and the Computer Society. The Computer Society needs to attack three areas: 1) The Society needs to be involved where jobs are produced, covering the practitioners' needs and promoting their participation, but linked to academia and local governments. 2) The Society must promote active international cooperation programs for encouraging new generations to enter the computer engineering world, throughout all regions. 3) The Society should focus on quality when products and services are delivered to its members to add value to the great products it produces, while international best practices must be implemented to support our valuable staff to improve our members' satisfaction, all together as a team.

Biography. José-Ignacio Castillo-Velázquez has worked for 17 years in the computer and telecommunication industries as a practitioner in the public and private sectors (including positions at TELMEX, COMSE, IFE, and DICI) as well as serving as a tenured professor in private and public universities (including UTM, UPAEP, and BUAP).

Castillo-Velázquez is a senior member of the IEEE, a member of the Computer Society, the IEEE Communications Society, and the Mexican Academy of Systems Sciences. He has been a tenured professor for telecommunications engineering at the University of Mexico City since 2008. He has authored 25 journal and conference papers, as well as 12 technical papers. Castillo-Velázquez has participated in more than 100 invited conferences as well as interviews for magazines, newsletters, radio, and TV. He received a BS in electronic sciences, with honors, and an MS in semiconductor devices, both from the Autonomous University of Puebla.

For the IEEE, Castillo-Velázquez has served on committees or boards of the following projects: E-Scientia project leader for Mexico, 2011; Computer Society Board of Governors member, 2011; Region 9 NoticIEEEro editor in chief, 2008-2011, Strategic Planning Committee member, 2010-2011; Virtual Regional Meeting Committee technical administrator, 2010-2011; Virtual Communities Ad-hoc Committee chair, 2008-2009, administrator, 2007.

Fred Douglis

Position statement. As a magazine editor, technical committee chair, conference organizer, and author, I have seen many aspects of the Computer Society. The Society is a nonprofit organization intended to help its members and disseminate information. At the same time, "nonprofit" does not equal "loss." At a time of increasing competition and economic hardship, it's important for the Society to sustain its revenues or cut costs.

The portfolio of publications and conferences has grown considerably; the Society therefore competes with itself, let alone other organizations. I want to be aggressive in identifying overlaps and consolidating the portfolio into a smaller set of key venues.

I also find great concern among my colleagues over the variation in publication policies. While similar organizations offer open access, the Society has difficulty attracting authors to volunteer their content under a strict copyright. I intend to move our copyright policies into the 21st century.

More at www.douglis.org/fred/BOG11.

Biography. Fred Douglis received a PhD in computer science from UC Berkeley. He has worked in industrial applied research throughout his career, including Matsushita, AT&T, IBM, and currently EMC. He also has been a visiting professor at VU Amsterdam and Princeton University. He received an IBM Outstanding Technical Achievement award for contributions to System S, productized as Infosphere Streams. His research interests include storage, distributed systems, and Internet tools and performance. He has published one book, 40 workshop or conference papers, seven journal or magazine articles, and more than 40 patents and patent applications.

Douglis has been an IEEE Computer Society volunteer since 1993 and became a senior member of IEEE in 1997. He was EIC of *IEEE Internet Computing* from 2007-2010 and has served on its editorial board since 1999. He formed the TC on the Internet, chairing it from 1997-2000, and previously chaired the TC on Operating Systems and Application Environments from 1996-1998. He chaired several steering committees; helped organize the first IEEE/IPSJ Symposium on Applications and the Internet (SAINT) in 2001, serving as program co-chair; and was general chair of the 1993 IEEE Workshop on Workstation Operation Systems. Douglis has been program chair of four major non-Computer Society conferences.

David S. Ebert

Position statement. With the rapid change in information access and technology ubiquity, the Computer Society must be agile and adapt to our membership's changing needs. We should target growth opportunities and the internationally changing IT professional demographic. We must maintain our core values and highest quality of our products and services, while innovating in their delivery. We need to have clear member value to attract and retain membership. I will foster this innovative excellence in several ways. First, we must actively recruit young members worldwide through new venues and engage them as active volunteers to provide creative and responsive products and services to our membership. Second, we need to actively explore new opportunities, utilizing new modalities, market opportunities, and the world-class innovative content from our conferences and journals. We need to adapt these technologies to transform the way we do business, making us leaders in technology-delivered material and services to our membership.

Biography. David Ebert has been actively engaged in Computer Society Board of Governors, conference, technical committee, and publications activities for more than 10 years. Ebert served as associate editor, associate editor in chief, and editor in chief for *IEEE Transactions on Visualization and Computer Graphics*, as an associate editor for *IEEE Computer Graphics and Applications*, and is a member of the management committee for *IEEE Transactions on Haptics*. He has been a member of the Computer Society Publications Board, a member of the Visualization and Graphics Technical Committee executive committee, and the ACM Siggraph Executive Committee. Ebert has been conference cochair, program cochair, and papers cochair of eight Society cosponsored conferences and heads the IEEE Visualization and Analytics Science and Technology Conference advisory board.

Ebert is the Silicon Valley Professor of Electrical and Computer Engineering at Purdue University, a university faculty scholar, and director of the Visual Analytics for Command Control and Interoperability Center, a US Department of Homeland Security center of excellence. He received a PhD in computer science from Ohio State University and performs research in visualization, visual analytics, and computer graphics. Ebert is an IEEE Fellow, a Golden Core member, and received the Meritorious Service Award for his Computer Society activities.

Hakan Erdogmus

Position statement. The publications industry has been undergoing major changes. Meanwhile, academic and professional communities have been striving to sustain themselves with new ways of collaboration and access to information. Our survival will depend on how successfully we continue to meet the evolving demands of our membership while exploring sustainable revenue streams and reconciling old and new ways of doing business. New media, value-added services, mobility, delivery speed, availability, easy searching, consolidation, cross-linking, openness, partnerships, and new pricing models will drive the Society's transformation in the years to come.

As a long-time volunteer with a proven track record in leadership roles and as a professional whose career has straddled both industry and academia, I have a sound perspective on our members' diverse needs, their communities, and the expectations of the next generation of computing professionals. If elected, I will use my skills and experience to help the Society overcome impending challenges.

Biography. Hakan Erdogmus is an independent consultant based in Canada, an adjunct faculty member at University of Calgary's Department of Computer Science, a senior member of IEEE, and an at-large member of the Society's Publications Board. Erdogmus served as editor in chief of *IEEE Software* from 2007 to 2010. In his role as EIC, Erdogmus worked with the staff and magazine boards to significantly expand the outreach of *IEEE Software* and implement new operational processes. In 2011, he received the Computer Society's Meritorious Service Award for these contributions.

Erdogmus specializes in software engineering practices, project management, and economics of software development. From 1995 to 2009, he worked as a research scientist at the Canadian National Research Council. In 2003, he received the Eugene L. Grant Award for best contribution in engineering economy from the American Society of Engineering. He helped organize more than 50 scientific conferences in various roles. Erdogmus received a PhD in telecommunications from Institut national de la recherche scientifique, Université du Québec; an MS from McGill University's School of Computer Science; and a BS from the computer engineering department at Bogaziçi University in Istanbul. For his full résumé, visit HakanErdogmus.net.

Hironori Kasahara

Position statement. The importance of computer technology has been increasing for safe, ecological, and comfortable lives due to the influence of technologies like mobile phones, games, PCs, cloud servers, supercomputers, automobiles, robotics, and medical systems. However, most IT-related societies, including our IEEE Computer Society, have been facing problems like declining subscriptions to periodicals and fewer members from industry. In addition, interest in computing technology among youth seems to be declining.

If elected, I will do my best to make the Computer Society more attractive to the members and people in industry and also to let the younger generation know about technological dreams. I have extensive previous and ongoing experience in research and development of green computing with industry and government, including low-power solar power-driven multicores, parallel software, and supercomputers to protect lives from natural disasters and cancer. My activity will also include support of CSDA certification for students. Please visit www.kasahara.cs.waseda.ac.jp/kasahara.html.en.

Biography. Hironori Kasahara served as a member of the Board of Governors from 2009-2011. He is chair of the IEEE Computer Society's Japan chapter, a board member of the IEEE Tokyo section, a member of the IEEE Japan Council Long-Term Strategy Committee, and a program committee member of many IEEE conferences including SC and PACT. Kasahara has chaired the Information Processing Society of Japan (IPSJ) SIG on computer architecture and served as vice program chair of the ENIAC 50th anniversary ICS. Kasahara received a PhD in electrical engineering from Waseda University, Tokyo, where he became an assistant professor in 1986 and an associate professor in 1988. He has been a professor of computer science since 1997 and is director of the Advanced Multicore Processor Research Institute.

Kasahara was a visiting scholar at the University of California, Berkeley, in 1985 and at the University of Illinois at Urbana-Champaign's Center for Supercomputing R&D from 1989 to 1990. Kasahara has received the IFAC World Congress Young Author Prize, IPSJ Sakai Special Research Award, and the Semiconductor Technology Academic Research Center Industry-Academia Cooperative Research Award. His works include 185 papers, 100 invited talks, and 400 news articles. He has led Japanese national projects on green computing, parallelizing compilers, and multicores in METI/NEDO.

Gargi Keeni

Position statement. As an IEEE Computer Society volunteer, I have always felt that the Society, as a flag bearer for computer and information technology professionals, has a larger role to play. Toward this end, it should have more penetration—in terms of reaching more people, more participation—by getting more people (men, women, students, researchers, and young professionals) to participate and more utility—by providing value for all members current and future. Above all, the Society must foster more creative ideas and activities. It should play a central role in bringing science, engineering, and human networking together to ensure better preparations for and responses to natural calamities.

As a member of the IEEE Computer Society Board of Governors, I will work toward these goals by leveraging my industry background and my existing roles in national and international forums. I will also work to forge closer ties with other professional bodies.

Biography. Gargi Keeni, a vice president at Tata Consultancy Services, has more than 20 years of multicultural and multilocation experience in software development and services delivery. Keeni is a senior member of IEEE and serves on the Computer Society's Industry Advisory Board. She has also served on the advisory board of *IEEE Software*.

Keeni received a PhD in physics from Tohoku University. She serves as a cochair of the advisory panel of the NASSCOM quality forum, leads the working group on life cycle and processes in ISO/PCS259, is a member of the business planning group SWG1 of ISO/IEC JTC1 SC7, and serves on the program committee of the IWFST. A certified examiner for JRD-QV, Keeni is an SEI-authorized instructor and lead appraiser for CMMI and was a lead appraiser for People CMM. Her interests include information security, process improvements, quality management systems, and business excellence.

Keeni is a member of the IEEE Calcutta section executive committee and chairs its professional activities committee. She is also a member of the Executive Committee of the Computer Society Chapter of Calcutta section. She was a founding member and vice chair of the IEEE Women in Engineering Calcutta section.

Fabrizio Lombardi

Position statement. The Computer Society's success is indeed the success of all of its constituencies; so the Society must continue to be inclusively collaborating with other professional organizations to proactively anticipate the needs of its members in education, technology, and the profession-at-large. My accomplishments in Society-sponsored publications and conference activities are proof of my capabilities. Through effective communication, proven leadership, and timely planning, I will contribute to the Board of Governors to help enable, facilitate, and expand compelling new initiatives to all members. I will help to enhance existing organizational structures and improve service and products.

Engaging volunteers, moving into new technical frontiers, enlarging student activities—these are few of the endeavors that I will be honored to pursue by fostering collegiate discussions and efficient resolutions. My background in technology, education, and management will add a further perspective in helping to shape our Society. Respectfully, I ask for your vote.

Biography. Fabrizio Lombardi holds the International Test Conference endowed chair at Northeastern University, Boston. He received a doctorate in electronic engineering from University College London. Lombardi is an IEEE Fellow and twice a Computer Society Distinguished Visitor. He has extensively contributed to Computer Society publications activities. Lombardi was twice the associate editor in chief and a two-term editor in chief of *IEEE Transactions on Computers* as well an associate editor of *ACM JETC* and *IEEE Design & Test of Computers*. Currently, he is an associate editor *of IEEE Transactions on Nanotechnology* and *IEEE Transactions on CAD of ICAS*. Within the Society, Lombardi has been an ex-officio member of the Publications Board, as well as the Operations Committees on Transactions and the Computer Society Digital Library. He serves as chair of the IEEE Computer Society Test Technology Technical Council Committee on Nanotechnology and is a member of the steering committee for the *IEEE Transactions on Nanotechnology*.

In professional activities, Lombardi has initiated and chaired many Society-sponsored meetings, such as the IEEE Symposia on NCA and DFT in VLSI Systems. His achievements have been recognized by awards from industry, the IEEE Engineering Foundation, and the Canadian, Japanese, and US governments.

Paolo Montuschi

Position statement. I feel that two issues have elevated priorities over other "consolidated" issues such as revenues, budget restrictions, and simplification of procedures.

The world of publications is experiencing rapid, technology-driven change. Fruition in mobility, frequent updates, podcasts, and communities are just the first steps toward new publishing models that should be aimed mostly at readers. Users should be provided with new and exciting reading experiences, and we should start thinking about and implementing new frameworks of "augmented reading" in which new technology-evolving benefits of e-publications are added to the classical reading experience.

The Computer Society, as a true international organization, must consider the needs of its members on a global scale. All members should be able to clearly understand and use what the Computer Society has to offer. This could be achieved by involving volunteers from different countries, by listening to the needs of members, and by taking action locally and centrally.

Biography. Paolo Montuschi is chair of the Control and Computer Engineering Department, chair of the board for financial external affairs, and a member of the Board of Governors at Polytechnic of Turin.

Montuschi serves as an associate editor of *IEEE Transactions on Computers*, chair of the Digital Library Operations Committee, and as a member of the Publications Board, the Electronic Publishing Services Committee, and the advisory board for Computing Now. For more than 20 years, Montuschi has been a member of the IEEE and the Computer Society, where he served as member-at-large of the Computer Society's Publications Board, member of an IEEE ad hoc committee for the quality of conference articles in IEEE Xplore, and member of the Conference Publications and Digital Library Operating Committees. He served as guest editor and associate editor of *IEEE Transactions on Computers* in 2000–2004, as well as program cochair and program committee member of several IEEE conferences.

Currently, within the Computer Society, Montuschi is actively involved in opening the door to new publication frameworks geared toward e-reading and mobile devices. He received an MS and a PhD in computer engineering and has been a full tenured professor since 2000.

Arnold N. Pears

Position statement. The IEEE Computer Society has a crucial role to play in the future of the computer science and engineering professions. It provides important standards and plays a major role in the education and professional development of future computing professionals. As a university professor with 20 years' experience in computing education and computing and engineering education research, I hope to contribute to strengthening the impact of the Computer Society's educational initiatives. As an active member of the Educational Activities Board, and having served as chair of the Steering Committee of the IEEE/ASEE Frontiers in Education conference series, I feel that the time has come to take a more active role in the policy-making process. My goal within the Board of Governors is to increase the impact of Computer Society educational initiatives on curricula that are relevant to the future of our profession.

Biography. Arnold Pears received a BS and a PhD from La Trobe University in Melbourne, Australia. He currently is an associate professor at Uppsala University, Sweden, where he leads the UpCERG research group in computing and engineering education, is a director of the CeTUSS national center for engineering education, is a member of the Uppsala University Academic Senate, and serves on the pedagogical advisory board to the Faculty of Science and Technology. Pears received the Uppsala University Pedagogy Prize in 1998. He has authored more than 40 articles and papers in major conferences and journals in computing and engineering education.

Pears has worked as a researcher in Australia, France, and Sweden during an academic career that spans 20 years. He served on the Dasher Awards Committee for the IEEE Frontiers in Education conference (2008-present), chairs the IEEE Nordic Education chapter, is a member of the IEEE Frontiers in Education conference series' Steering Committee (member 2010, chair 2011-12), serves on the committee of the new European EDUCON conference series, is a member of the Educational Activities Board (2010-present), and currently leads the effort to build up the STC in education.

William (Bill) Pitts

Position statement. I desire to bring my 25 years of conference experience to the Computer Society Board of Governors. I am committed to increasing conference surplus through tighter control over conferences with losses, while actively promoting the elimination of procedures that are obsolete or redundant with IEEE. As conference organizers, we need better software and systems for budgeting, paper submission, peer-review, proceedings publications, and closing to reduce the time it takes volunteers and staff to manage our conferences. With an improved infrastructure, we will be able to attract and select desirable conferences to sponsor financially. I championed the adoption of a reduced administration fee for large conferences and am a member of a committee working to do the same for very small conferences. I am committed to tightening the criteria and procedures for approving technically cosponsored conferences to insure quality and protect IEEE and Computer Society brands.

Biography. Bill Pitts currently serves as vice chair of the Technical and Conferences Activities Board, where he chairs the Technical Meeting Request Committee. TMRC is responsible for reviewing large conferences, any conference with a loss, and requests for technical cosponsorship. He is a member of the IEEE Conference Publications Committee, vice chair of the Technical Committee on Parallel Processing, and has served as founder and finance chair of the IEEE International Parallel & Distributed Processing Symposium (www.ipdps.org) for the past 25 years.

Previously, Pitts was a member of the IEEE Conferences Committee (2009–2010), chair of an ad hoc committee for developing the IEEE conference operations manual, finance chair of the Orange County section, and general chair and finance chair of the Orange County chapter of the Computer Society.

Pitts became an IEEE Computer Society Golden Core member in 2010, received the Society's 2010 Outstanding Contribution Award "for design and implementation of improved processes and systems for Technical & Conferences Activities Board, 2009-2010", and was awarded its 2010 Distinguished Service Award "for lifetime service to the International Parallel and Distributed Processing Symposium (IPDPS) community."

Sattupathu V. Sankaran

Position statement. The IEEE Computer Society has recorded its first year with growing membership in 2010, coinciding with the launch of the new Member and Geographic Activities Board, with intent to align with IEEE's MGA. This is also a juncture when ACM is launching initiatives in countries like China, India, and Russia. The Computer Society therefore must connect, better than ever before, with its members and potential members and prove its true value and benefits to them. This can happen by better engagement with the members and chapter leaders; better liaison with industry; higher quality publications, technical activities, and conferences; easier and more economical access to its digital library; and better standards and professional services. We should join hands in IEEE membership efforts. It will be my pleasure to continue to serve the Society.

Biography. Sankaran received a BS in electrical engineering from Jadavpur University, India, and an MS in control systems from the University of Pennsylvania. He has worked in industry for more than 30 years, including IBM and Bharat Heavy Electricals Limited in India, and Westinghouse, Electric Power Research Institute, and Duke Power in the US. Sankaran's interests focus on industry research, power plant training simulators, and industry-academic relations. He was senior professor and associate dean at the International Institute of Information Technology-Bangalore before moving on to corporate IT consulting, currently for SAP Labs.

Sankaran was a recipient of the Federation of Indian Chambers of Commerce and Industry award for outstanding industrial research in 1987 and the Society for Computer Simulation's Industry Technology Award for the EPRI Mobile Training Simulator in 1992.

A senior member of IEEE, Sankaran served in the IEEE Bangalore Section, rising to chair in 2002-2003. Sankaran served as membership development chair for Region 10 from 2004- 2006, and supported Computer Society headquarters over the years. He served the Computer Society as vice president of chapters activities in 2009 and as vice president of the new Member and Geographic Activities Board in 2010. He received the IEEE Millennium Medal and was named a Computer Society Golden Core member in 2010.

David J. Schultz

Position statement. I see several primary areas in which the Computer Society needs to evolve. First, we need to streamline our business model to become a more valuable and viable organization for the 21st century. We should reexamine our membership fee structure to ensure that we can cover our costs while remaining competitive. We also need to reevaluate our revenue model so that each service that we offer covers its own costs. Finally, we must become more effective in using Web-based and social networking communication channels, both to reach out and serve our current members and to recruit new members. Effective, results-oriented innovation must become the watchword of the Computer Society in both of these areas.

I look forward to the opportunity to work with the Board of Governors as it addresses these and other challenges.

Biography. David Schultz recently retired after a career with Computer Sciences Corporation for more than 25 years. His primary areas of expertise are software quality assurance and process improvement. He has developed, implemented, and managed software QA programs for a wide variety of projects. Most recently, Schultz was a software quality assurance lead on the Spectrum Management Transition Initiative project for ITT. Prior to this, Schultz served as a process improvement consultant for NASA's Goddard Space Flight Center.

Schultz received an MS in computer science from the University of Maryland. He is currently a member of the IEEE Computer Society Membership and Geographic Activities Board (MGAB), the Standards Activities Board (SAB), and the Educational Activities Board.

Schultz currently serves as vice chair for the MGAB Awards and Recognition Committee. During 2010, he chaired the MGAB Distinguished Visitor Program. Within the SAB, Schultz has served as a member of the Software and Systems Engineering Standards Committee Executive Committee since 1986 and currently chairs its management board. He also serves on the IEEE Computer Society Awards Committee, for which he chairs the Hans Karlsson Award subcommittee. Schultz has been a member of the Computer Society Golden Core since 1996.

IEEE COMPUTER SOCIETY ELECTION

Cast your vote quickly and easily on the Web at www.computer.org/election2011. Ballots must be received no later than 12:00 noon EDT on Tuesday, **4 October 2011**. To vote by mail, use the return-mail envelope provided and send your ballot to the address provided below.

Return ballots by mail to:
IEEE Computer Society
c/o Survey & Ballot Systems
PO Box 46430, Eden Prairie, MN 55344, USA

IEEE President-Elect Candidates Address Computer Society Concerns

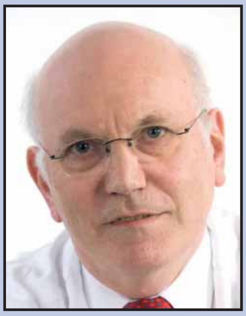
The IEEE Computer Society has established a reputation for excellence within the fields of computing. As a component of the IEEE, the Computer Society's activities parallel those of 38 other IEEE societies and councils serving the computing and engineering disciplines. Representing by far the largest IEEE society contingent, the Computer Society has 85,000 members, approximately 60 percent of whom are full IEEE members.

Recognizing the impact of IEEE leadership over the Computer Society and in turn the power of Computer Society members' votes to influence the selection of the IEEE leadership, we posed questions to this year's IEEE president-elect candidates. Because this election determines who will serve as president-elect in 2012, president in 2013, and past president in 2014—vital positions within the IEEE's governing body—our members must cast informed votes.

Our volunteer leaders have identified the following questions as essential to the Computer Society, IEEE, and the Computer Society's relationship with IEEE. The first response to each question states the Computer Society's position. These positions synthesize the views of our most senior leadership: the Society's current, past, and incoming presidents. We present these questions and answers (limited to 150 words each) to help you make your decision in the IEEE annual election. Only ballots received by noon, central time, on **3 October 2011** will be counted.

We also remind and encourage you to cast your vote for Computer Society leaders by **4 October 2011** in our Society election.

—John Walz, IEEE Computer Society President-Elect



ROGER D. POLLARD

Roger Pollard's professional career has been in both academia and industry. Until September 2010, he was Professor and Dean of Engineering at the University of Leeds, UK. Since 1981, he has also been a consulting engineer at Agilent Technologies (formerly Hewlett-Packard Company), Santa Rosa, California. He has made research contributions to high-frequency devices and is noted for work in microwave and millimeter-wave circuits, network measurements, calibration, error correction, and large-signal characterization.

Pollard has authored contributions in books, more than 150 journal articles, three patents, and commercial publications. He was elected to the UK's national academy, the Royal Academy of Engineering, in 2005, and is an IEEE Fellow, a chartered engineer, and a Fellow of the IET (formerly IEE). Contact him at www.rogerpollard.org.

PETER W. STAECKER

Peter Staecker received BS and EE degrees from MIT and an MS and PhD from Polytechnic University. His professional career started in 1972 at MIT Lincoln Laboratory, where he developed microwave design and test techniques for satellite communications. In 1986, Staecker joined M/A-COM, where he led program, product, and process development, then helped the company's transition from defense to commercial markets. During this period, he also established strong ties with US and European universities and with research organizations. He retired from M/A-COM as Director of Research & Development. Staecker has served industry and government on manufacturing and advisory panels and is a consulting editor to *Microwave Journal*. He is the past-president and an Honorary Life Member of the MTT Society and is an IEEE Life Fellow. Staecker's 28-year service to IEEE includes leadership roles in finance, strategic planning, publications, and membership. He has served on the IEEE Board of Directors for five years.

QUESTION 1: Emerging technologies

1 An increasing number of emerging technologies are competitors to long-established groups. In attempting to present an interdisciplinary face to the world, we often foster internal fragmentation that may threaten our effectiveness.

- How would you integrate emerging technologies into the current IEEE structure?
- How would you encourage new groups to cooperate with or be adopted by existing units?
- How can the IEEE encourage and support the inclusion of new technologies without diverting support from the existing entities?

COMPUTER SOCIETY POSITION

1 The president should lead the IEEE in developing effective mechanisms to determine the viability of new groups and then help viable groups integrate into existing entities. While IEEE funding has jump-started smart grid, cloud computing, life sciences, and other new initiatives, IEEE is generally late to the game by recognizing emerging technologies.

One effective mechanism to recognize emerging technologies is by analyzing Xplore downloads for spikes from subscribers with broad interests across several Societies' field of interest. After funding an incubation period to organize leaders from several Societies/Councils, a Multisociety Technical Group can be organized under TAB. This structure could encourage the inclusion of new technologies without diverting support from existing entities. It could also attract and expand the audience of authors and subscribers by defining their field of interest using the Xplore indexing scheme. Finally, virtual journals in focused areas of interest can be created.

Pollard

1 With changes in technology happening so quickly and frequently, IEEE's biggest challenge is keeping pace with the creation of member communities in emerging technologies. We need to be agile and flexible in welcoming as members people working in areas that are outside our present comfort zones.

We must recognize that today's IEEE is divided into disciplines, but it is the multidisciplinary problems that define the real issues facing society. These problems have been expressed in terms of the "grand challenges," all of which are interdisciplinary and most sit in IEEE fields of interest. To support members operating in multidisciplinary technology, we need to create new multidisciplinary technical communities comprising member societies to open up these new areas. The Computer Society is fortunate in that its subject is an essential part of the majority of solutions and it has the benefit of being embedded in the technically rich environment of IEEE.

Staecker

1 First, follow the lead of the Technical Activities' Future Directions Committee (FDC) which currently

- identifies technology trends and coverage opportunities, and
- fosters development of products and services in these areas.

The goal is to create a model for nurturing a community within Technical Activities' current structure that reaches across existing societies while maximizing member and customer value with minimal governance and bureaucracy. The FDC has been encouraged to develop a "maturity model," and I support this effort. This model should also periodically examine common technical fields of interest among communities in neighboring fields of interest, with the view of better serving the member, customer, and public.

Encouraging new technologies through new conferences or special topics publications will add relevance to membership and customers, while providing identity and support to contributors. Colocation of these conferences can support the new technology within the structure of the parent community

QUESTION 2: Professional development

2 An important role and contribution of IEEE to industry is to improve the caliber of professionals, whether currently employed or about to enter the work force. Professional development activities and continuing education are valued by industry members and their managers and are key to membership retention. However, IEEE has offered limited products for continuing education, self-directed learning, and credentialing.

- In your opinion, how much importance should IEEE place on professional development for its members?
- How can the IEEE help its societies offer professional development products and services?

COMPUTER SOCIETY POSITION

2 The Computer Society has offered self-directed learning as a member benefit, along with certifications and professional education programs in software engineering.

IEEE can help Societies like the Computer Society by providing resources for market research, sales and marketing, and funding for new initiatives aimed at professional development. We support the expansion of the all-IEEE professional education eLearning platform as the vehicle for high-quality asynchronous learning. This product needs to respond to market trends and demonstrate a good balance between breadth and depth. The IEEE president can support these efforts by making career development and lifelong learning a priority for his presidential year.

Pollard

2 It is clear from surveys that the number one reason members join IEEE is to remain technically current. Members also tell us that they see membership as enhancing career opportunities and providing continuing education. Therefore, IEEE needs to invest in professional development products, improving the quality, availability, marketing, and support. We need a central professional IEEE facility for the creation of multidisciplinary continuing education products, making it easy for Society volunteers to contribute to the creation of high-quality materials. Continuing education needs to be offered as a member benefit and have a coherent structure, not just a menu of modules, that offers a recognized qualification on completion.

IEEE should not compete with commercial activities or with our own members who are professionally engaged in providing CE, but buy in services as appropriate.

Staecker

2 Professional development is one of IEEE's most important services to its members because it includes both soft and technical skills necessary for career success.

At the top level, I would encourage Educational Activities, Standards Activities, and Technical Activities to work together to formulate needs for the practicing technologist while identifying best service practices. They should also develop affordable and accessible common infrastructure that societies and sections can use. Societies with existing programs should be prominent members of this task force. IEEE marketing and research should work with societies without programs to determine what programs would be effective for members—and nonmembers—in their technical space. Sections and their chapters should be engaged to advise and execute efforts to create workshops or short courses. This is truly a cross-disciplinary effort, and one that is of greatest impact when it is relevant at the local level.

REPORT TO MEMBERS

QUESTION 3: Industry and academia

3 Over the past five years, the Computer Society has been aggressive in extending its reach into industry. In some cases, we have found it to be a challenge to connect these initiatives to our base of volunteers in academia.

- What IEEE experience do you have in developing programs that connect the two communities?

COMPUTER SOCIETY POSITION

3 Technology transfer between research and theory and commercial applications should be the sweet spot for IEEE and the Societies/Councils to occupy, as evidenced by the venture capital industry. The Computer Society's Industry Advisory Board's Industry Perspective papers could be posted in a separate area in Xplore for access by practitioners. Furthermore, requesting conference authors to supplement their abstracts with their view of the paper's relevance to industry could assist in the selection of sets of conference papers that can be commercialized.

Pollard

3 I believe that it is important that industry and academia must not be seen as being in competition; both contribute in different ways to the richness of modern technology. Many academics also work in industry—we must identify these people and use them to forge stronger bonds with companies. As well as carrying out research that finds its way into commercial products, academics recognize that they are training their students to work in industry—this mutual interest is a good point of contact that IEEE must understand and foster through activities in student branches. We must recognize that IEEE conferences are often the best venue to form links. The combination of an academic conference with a trade show will ensure that there are plenty of opportunities for the two communities to mix. I have more than 30 years of experience with successful programs connecting academia and industry.

Staecker

3 First, congratulations on establishing your Industry Advisory Board. These external industrial leaders are telling us how to be relevant to our practitioner community, and they represent a model that should be replicated throughout IEEE. Similarly, a Computer Society Academic Advisory Board can advise the Society on creating value for the academic community; in addition, having a joint meeting of the two might bring academics and industry folks to identify a single purpose on broader issues that are important to both groups.

In IEEE, my experience with Technical Activities as well as my own Society (MTT) has been that the key to connecting everyone is the Strategic Plan. First, have one, and then make sure one of your strategies is to serve the two communities or, better yet, connect them.

QUESTION 4: Infrastructure growth

4 How would you ensure that IEEE follows prudent financial management practices and prevents excessive growth of the infrastructure?

- What investments should be made now to support future IEEE growth?

COMPUTER SOCIETY POSITION

4 The Computer Society supports IEEE's plans to modernize its IT infrastructure, which should reduce staff efforts and allow volunteer leaders easier and quicker access to tailored and relevant business and technical information. On the other hand, IT centralization can lead to a "one size meets all" approach that may stifle the operating units' ability to serve specialized markets. A service-oriented architecture can produce software for creating and using business processes, packaged as services for every OU, which have varying needs, to tailor and subscribe for their growth.

Pollard

4 It is vital that IEEE has an efficient, effective, coherent, flexible, responsive, affordable, and consistent infrastructure. Economies of scale can be derived from doing as much as possible centrally. But we must work to determine the right level of central costs and how to share fairly these expenses. We must be sensitive to the different needs of those parts of our organization that operate in many businesses and vary the rules accordingly.

The guidelines for IEEE's financial management must ensure that spending proposals have good business reasons, are financially sound, and are properly reviewed. Financial transparency is vital, and volunteers throughout IEEE's organization should understand how IEEE finances work and contribute to the decision-making. We must invest in quality IT systems that not only provide access to IEEE's world-class information resources but also improve support for members and customers worldwide.

Staecker

4 The Executive Director has accountability for the IEEE staff. Head count has remained relatively flat for many years—in spite of our growth in revenue. This piece of IEEE infrastructure is not the problem.

We should identify appropriate metrics at the organizational unit level and monitor them by reporting them up through the OU treasurers to the IEEE Financial Committee for disclosure to the IEEE Board of Directors.

To support future IEEE growth, the BoD has approved investments in IT and publications infrastructure and new initiatives at a rate exceeding 5 percent of revenue per year. This is a necessary commitment to growth.

In addition, we should

- consolidate efforts in the publishing and conference space, both in-house and with external vendors, to achieve economies of scale;
- encourage opportunities for growth in our new initiative portfolio; and,
- examine near-term strategic objectives for insight into other areas of smart spending.

REPORT TO MEMBERS

QUESTION 5: Conferences

5 As IEEE's conference program has grown, so has the need to develop improved oversight for finances and quality. However, these requirements have resulted in increased work for conference organizers.

- What do you think the appropriate balance should be?

COMPUTER SOCIETY POSITION

5 Every IEEE control needs a corresponding automation strategy to provide a balance to volunteer leaders. Every IEEE control has a measured cost in terms of staff and volunteers' time. These cost needs to be invested into modern software tools for automation to attract and retain world-class conferences.

Pollard

5 The strong growth in conferences in recent years has resulted in some meetings in which material of questionable quality has been included. This tarnishes the overall IEEE brand and has a negative effect on the reputation of the whole of IEEE output. This problem, and the effects of having more conferences often organized by less-experienced people, has led to the need for increased oversight of the management of IEEE's conference business. Volunteer conference organizers guard their independence and are, rightly, concerned about the increased workload brought about by the need for improved accountability.

To achieve an appropriate balance, we need clear guidelines for potential conference organizers on the requirements necessary to ensure financial management and technical quality. IEEE must provide a comprehensive set of easy-to-use tools for conference organizers that will significantly simplify their tasks and help them ensure compliance with IEEE requirements.

Staecker

5 Quality includes relevance to the target audience, financial return, technical excellence, and volunteer recognition. The proper balance is a matter of culture and vision—of the event, its location, the sponsoring organization(s), and the volunteer leaders.

A hands-on culture puts a premium on

- value added to the volunteer experience,
- direct feedback on quality issues via interaction between conference volunteers and the "customers," and
- increased positive financial return.

The key to balancing the increased reporting needs while keeping volunteer work manageable is to create effective management and accounting tools that allow most of the data to be recorded and reported automatically. We need to put in the hands of conference organizers a suite of tested and user-friendly tools that would allow them to focus on their role as intellectual leaders of the activity and minimize the need to provide routine reports and financial results.

CALLS FOR ARTICLES FOR *COMPUTER*

Computer seeks submissions for an April 2012 special issue on interaction beyond the keyboard.

Interaction with computers has become an integral part of daily life for most people. As computing technologies proliferate, simple user interfaces and ease of use become key success factors for a wide range of products.

Although the keyboard and mouse are still the dominant user interfaces in home and office environments, with the massive increase in mobile device usage and the many new interaction technologies available, the way we interact with computers is becoming richer and more diverse. Touch-enabled surfaces, natural gestures, implicit interaction, and tangible user interfaces mark some of these trends.

Computer seeks original research that describes groundbreaking new devices, methods, and approaches to human-computer interaction in a world of ubiquitous computer use. Suitable topics include interactive surfaces and tabletop computing; tangible interaction and graspable user interfaces; and user interfaces based on physiological sensors and actuators.

Direct inquiries to guest editor Albrecht Schmidt of the University of Stuttgart at albrecht@computer.org.

Paper submissions are due **1 November**. For author guidelines and information on the electronic submission process, visit www.computer.org/portal/web/peerreviewmagazines/computer.

Computer seeks submissions for a September 2012 special issue on modeling and simulation of smart and green computing systems.

Sustainable and efficient utilization of available energy resources is perhaps the fundamental challenge of the current century. Academic and industrial communities have invested significant resources in developing new solutions to address energy-efficiency challenges in several areas including IT and telecommunications, green buildings and cities, and the smart grid.

Modeling and simulation methodologies are necessary for the comprehensive performance evaluation that precedes costly prototyping activities for such complex, large-scale systems. This special issue aims to disseminate the latest advances in modeling and simulation of smart and green computing systems, which are critical from the perspective of sustainable economic growth and environmental conservation.

Topics of interest include modeling and simulations of energy-efficient computing systems, green communications systems, and smart grid applications. For author guidelines and information on how to submit a manuscript electronically, visit www.computer.org/portal/web/peerreviewmagazines/computer.

Articles are due by **1 March 2012**. Visit www.computer.org/portal/web/computingnow/cocfp9 to view the complete call for papers.

CALLS FOR ARTICLES FOR IEEE CS PUBLICATIONS

IEEE Internet Computing plans a May/June 2012 special issue on infrastructures for online social networking services.

The proliferation of rich social media, online communities, and collectively produced knowledge resources has accelerated the convergence of technological and social networks, resulting in a dynamic ecosystem of online social networking services, environments, and applications. OSN sites' success is reshaping the Internet's structure, design, and utility. Moreover, this trend is creating numerous challenges and opportunities for the development, deployment, management, and operation of scalable, secure, interoperable OSN infrastructures that can sustain a cycle of innovative application development, improved end-user experience, high-quality service provision, privacy protection, and healthy market expansion.

IC's guest editors seek recent research results in systems, soft-

SUBMISSION INSTRUCTIONS

The Call and Calendar section lists conferences, symposia, and workshops that the IEEE Computer Society sponsors or cooperates in presenting.

Visit www.computer.org/conferences for instructions on how to submit conference or call listings as well as a more complete listing of upcoming computer-related conferences.

0018-9162/11/$26.00 © 2011 IEEE

EVENTS IN 2011

September

11-14 .... ASAP 2011
14-16 .... ECOWS 2011
18-23 .... ITC 2011
25 .... ICSM 2011
26-30 .... Cluster 2011

October

4-7 .... LCN 2011
4-7 .... SRDS 2011
10-14 .... PACT 2011
22-25 .... FOCS 2011

November

6-10 .... ASE 2011
6-13 .... ICCV 2011
7-9 .... ICTAI 2011
12-18 .... SC 2011
21-23 .... NCCA 2011

NCCA 2011

The goal of the First IEEE Symposium on Network Cloud Computing and Applications is to provide a forum for researchers and practitioners to discuss all aspects of network cloud computing. Conference organizers have solicited papers on topics that include self-healing cloud computing platforms, application exchange modules, authentication strategies, and dynamic reconfiguration of services. Work-in-progress papers, industry papers, and student papers are also encouraged.

NCCA 2011 takes place 21-23 November in Toulouse, France. Visit http://sites.google.com/site/ieeencca2011 for complete conference details.

ware, and services that provide novel ubiquitous, scalable, secure, and trustworthy OSN infrastructures.

Articles are due by **1 September**. Visit www.computer.org/portal/web/computingnow/iccfp3 to view the complete call for papers.

Computing in Science & Engineering plans a May/June 2012 special issue on scientific computing with graphics processing units.

GPUs aren't just for graphics anymore. These high-performance many-core processors are used to accelerate a wide range of science and engineering applications, in many cases offering dramatically increased performance compared to CPUs. Computer architects also use them to build the world's largest supercomputers. However, the use of GPUs in scientific computing comes with added risks. The effort needed to port applications can be substantial, and not every application benefits equally well from GPU acceleration.

The guest editors seek contributions covering all aspects of using GPUs to solve challenging computational science problems. Of special interest are articles presenting the results of porting efforts of large-scale scientific applications on large-scale GPU-based high-performance computers.

Articles are due by **14 September**. Visit www.computer.org/portal/web/computingnow/cscfp3 to view the complete call for papers.

CALENDAR

SEPTEMBER 2011

11-14 Sept: ASAP 2011, 22nd IEEE Int'l Conf. on Application-Specific Systems, Architectures, and Processors, Santa Monica, California; http://asap-conference.org

14-16 Sept: ECOWS 2011, 9th IEEE European Conf. on Web Services, Lugano, Switzerland; http://ecows2011.inf.usi.ch

18-23 Sept: ITC 2011, Int'l Test Conf., Anaheim, California; www.itctestweek.org

25 Sept-1 Oct: ICSM 2011, 27th IEEE Int'l Conf. on Software Maintenance, Williamsburg, Virginia; www.cs.wm.edu/icsm2011

26-30 Sept: Cluster 2011, IEEE Int'l Conf. on Cluster Computing, Austin, Texas; www.tacc.utexas.edu/ieee/2011/index.php

OCTOBER 2011

4-7 Oct: LCN 2011, 36th IEEE Conf. on Local Computer Networks, Bonn, Germany; www.ieeelcn.org/index.html

4-7 Oct: SRDS 2011, 30th IEEE Int'l Symp. on Reliable Distributed Systems, Madrid, Spain; http://lsd.ls.fi.upm.es/srds2011

10-14 Oct: PACT 2011, 20th Int'l Conf. on Parallel Architectures and Compilation Techniques, Galveston, Texas; http://pactconf.org

22-25 Oct: FOCS 2011, 52nd IEEE Symp. on Foundations of Computer Science, Palm Springs, California; www.cs.ucr.edu/~marek/FOCS11

NOVEMBER 2011

6-10 Nov: ASE 2011, 26th IEEE/ACM Int'l Conf. on Automated Software Eng., Lawrence, Kansas; www.continuinged.ku.edu/programs/ase

6-13 Nov: ICCV 2011, 13th Int'l Conf. on Computer Vision, Barcelona, Spain; www.iccv2011.org

7-9 Nov: ICTAI 2011, 23rd IEEE Int'l Conf. on Tools with Artificial Intelligence, Boca Raton, Florida; www.cse.fau.edu/ictai2011

12-18 Nov: SC 2011, ACM/IEEE Int'l Conf. for High Performance Computing, Networking, Storage, and Analysis, Seattle; http://sc11.supercomputing.org

21-23 Nov: NCCA 2011, First IEEE Symp. on Network Cloud Computing and Applications, Toulouse, France; http://sites.google.com/site/ieeencca2011

DECEMBER 2011

5-8 Dec: E-Science 2011, 7th Int'l Conf. on e-Science, Stockholm, www.escience2011.org

ADVERTISER INFORMATION • AUGUST 2011

ADVERTISER	PAGE
Apple	90, 92, 99
Applied Materials, Inc,	86
Broadcom Corp.	84, 86, 89, 106, 109
Cisco	89
Composite Software, Inc.	87
Hong Kong Polytechnic University	89
HP Enterprise Services, LLC	85, 88
IEEE CSDP	Cover 2
IEEE CS Press	53
Institute for Defense Analyses	84
Juniper Networks	85
Nokia	109
Technical University of Denmark	87
University of Maryland University College	64
Classified Advertising	84-92

ADVERTISING PERSONNEL

MARIAN ANDERSON: SR. ADVERTISING COORDINATOR
Email: manderson@computer.org; P: +1 714 821 8380; F: +1 714 821 4010
Sandy Brown: Sr. Business Development Mgr.
Email sbrown@computer.org; P: +1 714 821 8380; F: +1 714 821 4010

ADVERTISING SALES REPRESENTATIVES
(DISPLAY ADS)

Western US/Pacific/Far East:
Eric Kincaid
Email: e.kincaid@computer.org
Phone: +1 214 673 3742
Fax: +1 888 886 8599

Eastern US/Europe/Middle East:
Ann & David Schissler
Email: a.schissler@computer.org, d.schissler@computer.org
Phone: +1 508 394 4026
Fax: +1 508 394 4926

ADVERTISING SALES REPRESENTATIVES
(CLASSIFIED LINE/JOBS BOARD)

Greg Barbash
Email: g.barbash@computer.org
Phone: +1 914 944 0940
Fax: +1 508 394 4926

CAREER OPPORTUNITIES

TEXAS STATE UNIVERSITY-SAN MARCOS, Assistant Professor in Software Engineering. Applications are invited for a tenure-track position at the rank of Assistant Professor. Applicants must have completed all requirements for a PhD with specialization in software engineering by start of employment. Consult the department recruiting page at http://www.cs.txstate.edu/recruitment/faculty_recruit.php for job duties, qualifications, application procedures, and information about the university and the department. Texas State University-San Marcos will not discriminate against any person in employment or exclude any person from participating in or receiving the benefits of any of its activities or programs on any basis prohibited by law, including race, color, age, national origin, religion, sex, disability, veterans' status, or on the basis of sexual orientation. Texas State University-San Marcos is a member of the Texas State University System.

ADS ALLIANCE DATA SYSTEMS, INC. has a position in Arlington, VA: Software Engineer: Exp. in software; load & performance testing, stress testing, bug tracking, integration testing, functionality testing, & smoke testing; software functional testing of web-based ASP.Net apps using C# & VB.NET technologies; writing integration tests; & other duties/skills required. [Job ID# AD-11VA-SW]. Mail resumes to S. Resler-HR, Alliance Data Systems, 601 Edgewater Dr, Wakefield, MA 01880 & note Job ID#.

BUSINESS APPLICATIONS DEVELOPER (Dreyer's Grand Ice Cream, Inc., Bakersfield, CA): As team lead, engage in research, design, & devel. of comp. soft. sys. in conjunction w/hardware product devel. Req. Bach's deg. in Comp. Sci., Elec. Eng., Math., Bus. Admin., or rel. field & 5 yrs' exp. in research, design, & devel. of comp. soft. sys. in conjunction w/hardware product devel. All stated exp. must incl. the following: PL/SQL; Perl scripting; app. of Elec. Data Interchange ("EDI") principles & best practices; analysis of soft. reqs. to determine design feasibility; devel. & formulation of comp. sys.; devel. & maint. of project doc.; & acting as team lead for devel. & bus. analysts. 5 yrs' of stated or other exp. must incl. SAP ABAP/4. Exp. may, but need not, be gained concurrently. Apply w/ resume to M. Simo, Nestle USA, Inc., 800 N. Brand Blvd., Glendale, CA 91203. Ref. job code BAD. NO PHONE CALLS OR THIRD PARTY AGENTS PLEASE.

DELL SERVICES ENGINEERING SOLUTIONS CORP. has an opening in Peoria, IL for a Software Development Engineer to perform embedded software design, development & validation for heavy engineering/ off-road vehicle electronic controls systems. Requires Master's or Bachelor's + experience. Send resumes to PSC-PRMResume@ps.net. Job code 110019UF must be referenced in email subject line. EOE.

COMPUTER & INFORMATION SYSTEMS MANAGER/DIRECTOR - Plan, coord, & direct research on comp-related activities; help determine goals of an organization & then implmt technology to meet those goals; oversee all tech'l aspect of an organization, such as s/ware dvlpmt, n/work security, & Internet operations; direct work of other IT professionals, such as comp s/ware engrs & comp support specialists; plan & coord activities such as installing & upgrading h/ware & s/ware, prgmg & systems dsgn, implmtn of comp n/works, & dvlpmt of Internet & intranet sites; upkeep,

Positions at the Institute for Defense Analyses Center for Computing Sciences

The Institute for Defense Analyses Center for Computing Sciences (IDA/CCS) is looking for outstanding researchers to address difficult computing problems vital to the nation's security. IDA/CCS is an independent, applied research center sponsored by the National Security Agency (NSA). Emphasis areas for IDA/CCS technical staff include high-performance computing, cryptography, and network security. Members of the technical staff come from a diverse variety of backgrounds, including computer science, computer architecture, computer/electrical engineering, information processing, and the mathematical sciences; most have Ph.D.s. Special attention is paid to the design, prototyping, evaluation, and effective use of new computational algorithms, tools, paradigms, and hardware directly relevant to the NSA mission. Stable funding provides for a vibrant research environment, and an atmosphere of intellectual inquiry free of administrative burdens.

The center is equipped with a very large variety of hardware and software. The latest developments in high-end computing are heavily used and projects routinely challenge the capability of the most advanced algorithms and architectures. IDA/CCS research staff members have always been at the forefront of computing, as evidenced by lasting, visible contributions to areas as varied as multi-threaded architectures (e.g., Horizon), novel computing systems (e.g., FPGA-based Splash and Splash-2, Processing-In-Memory chips), design and implementation of operating systems (e.g., the Linux kernel), and programming language design and implementation for high-performance computing systems (e.g., Universal Parallel C and Cinquecento).

IDA/CCS research staff work on complex topics often engaging multidisciplinary teams; candidates should demonstrate depth in a particular field as well as a broad understanding of computational issues and technology. Because the problems of interest are continually evolving, IDA/CCS recruitment focuses on self-motivation, strength of background, and talent, rather than specific expertise.

Located in a modern research park in the Maryland suburbs of Washington, DC, IDA/CCS offers a competitive salary, an excellent benefits package, and a superior professional working environment. U.S. citizenship and a Department of Defense TSSI clearance (with polygraph) are required. IDA/CCS will sponsor this clearance for those selected. The Institute for Defense Analyses is proud to be an equal opportunity employer.

Please send responses or inquiries to:

Dawn Porter
Administrative Manager
IDA Center for Computing Sciences
17100 Science Drive
Bowie, MD 20715-4300
dawn@super.org

Broadcom Corp.

is seeking a

Scientist, Staff II – Design

Irvine, CA
Ref: ENG6-IRCAJH

Reqs MS in EE & 3 yrs exp. Reqs Cadence Virtuoso schematic entry & layout, Cadence Analog Artist & Spectre, Calibre Verification, data analysis of test silicon & SoC products & debuging and micro-probing of integrated analog circuits. May travel up to 5% to attend project meetings (travel will be domestic/international).

Mail resumes to:

HR Operations Coordinator
5300 California Ave.
Bldg. 4, #42069
Irvine, CA 92617
Must reference
job code ENG6-IRCAJH.

Published by the IEEE Computer Society

maintenance, & security of n/work; analyze comp & inform. needs of firm from an operational & strategic perspective & determine immediate & long-range personal & eqpmt req.; assign & review work of subordinates & stay abreast of latest technology to ensure that the firm remains competitive. Must have avlble refs., either MBA w/technology as core component or its US equiv or B.A. deg in comp related field or its US equiv, + min of 5 yrs. of exp as IT Mgr/Director. Mail resumes to: TravelAllRussia LLC, Attn: Ms. Pakhomova, 2300 N Pershing Dr., Ste 202, Arlington, VA 22201.

NOKIA INC. has a position in San Diego, CA: Senior Audio Designer Engineer: Exp. involving audio/acoustics design of mobile phone, including transducer selection, acoustic simulation, acoustics chamber design, audio schematics/layout design & other duties/skills required. [Job ID: NOK-SD11-SADE]. Mail resume to Nokia Recruiter, 3575 Lone Star Circle, Ste. 434, Ft. Worth, TX 76177 & note Job ID.

DATA ANALYST w/BackOffice Assoc. (S. Harwich, MA). Participate in data migration by developing product req'ts. Analyze & evaluate data processing/manipulation sys. Need Master's w/knowledge of SQL query design/database bldg., SAP modules & data mining. Will also require travel to client sites. See www.boaweb.com/careers.htm to complete online app. & upload resume (resume only pls. - 1 doc. limit on the upload). Reference "Job Code KVY" when applying.

DATABASE ADMINISTRATOR, wanted by Japanese Software Consulting Co. in Berkeley. Dsgng web-based system, d/base, & applic. Must have BA in Info Systems, Comp Sci, S/ware Engg or rltd fld. Superior bilingual abilities in Japanese & English. Send resume to: Kamiya Consulting Inc., 1913 Addison St., Ste. 303, Berkeley, CA 94704.

PRINCIPAL SOFTWARE CONSULTANTS sought by Nexius Solutions, a leading Wireless Technology Consulting firm w/comprehensive suite of services designed to meet business & tech needs of wireless operators & enterprises, in Allen, TX. Responsible for evaluating & recommending s/ware tech choices for wireless & telecomms n/works. To identify product & technical reqmts & work w/potential suppliers to determine how they meet customer objectives. To work within team of technical experts in next generation mgmt technologies through long term customer engagements. Min. req Masters Industrial & Systems Eng or rltd deg. Send resumes to 825 Market St., Ste. 250, Allen, TX 75013.

VP, SOLUTION SALES (Islandia, NY). Establish & control corp sales strategy & force; Ensure mgmnt delivers solutions that address client's bus reqs; Run $30+ Million SaaS bus. & direct & lead sales staff. Direct, manage & lead teams in Sales; reinforce Software Sales Models & operations; Lead & direct teams in Solution Sales; Manage region-wide opps & transactions & analyze fin'l & perf metrics in US & European market in context of multinational corp. Reqs: Master's deg or for. equiv in CS, Bus. Admin, MIS, Eng (any) or rel. tech + 2 yr exp in job offd &/or rel, Employer will accept Bach's deg or for. equiv. in CS, Bus. Admn., MIS, Eng (any) or rel. tech + 5 yrs prog exp in job offd &/or rel. Must have exp. w/: Software as a Service (SaaS) bus. model & market principles; Running a $30 Million SaaS bus.; Leading teams in Sales or customer facing setting; European market in multinational corp; Software Sales Models & Operations. Frequent travel reqd; work from home benefit available. Send resume to: Althea Wilson, CA Technologies, Inc., One CA Plaza, Islandia, NY 11749, Requisition #23625.

SYSTEMS ANALYST - dsgn, dvlp, test & implmt applic s/w using Perl, Mod_Perl, CGI, Apache 2.x, HTML, JQUERY, XML, JSON, RDBMS (Sybase/Oracle/mysql), Korn Shell, LDAP, Autosys, Web Sphere 4.0/5.0, Unix/Solaris, Core Java, Windows XP/NT/98/95, MS Office 2003, RH Linux, 6.0/7.0/8.0/9.0, Solaris. LDAP, DBArtisan, Version control, Web Services preferred. Frequent travel reqd. Reqs BS Comp Sci, Eng or rel w/5 yrs exp. Mail resumes to Astir IT Solutions Inc., 50 Cragwood Rd., Ste. # 219, South Plainfield, NJ 07080-2433.

NOKIA INC. has a position in Sunnyvale, CA: Senior Software Engineer: Design & develop multimedia system software for Symbian Smartphones, work with Series 60 or Symbian RTOS for mobile platforms in software design & development & C++ in UNIX environment, video & audio codec, multimedia technologies, & other duties/skills required. [Job ID: NOK-SV11S-SRE]. Mail resumes to Nokia Recruiter, 3575 Lone Star Cir, Ste. 434, Ft. Worth, TX 76177 & note Job ID#.

CITCO TECHNOLOGY MANAGEMENT INC. has an opening in Jersey City, NJ for Application Developer to support, enhance & develop Oracle database applications for Citco Financial Products & Citco Transparency Platform business streams. Send resumes to employment@citco.com. Please reference Job code CTM54 in email subject line.

NOKIA SIEMENS NETWORKS US LLC (NSN) has the following positions in Irving, TX: Converged Core Engineer: Troubleshoot & implement telecom switches; support mobile core circuit technologies & provide technical support for customer networks; responsible for the overall performance of core & applications elements; & other duties/skills as required. [Job ID: NSN-11STX-CCE]. System Integration Engineer: Design, develop & test telecom protocols; integrate various network elements & third party products to provide customized customer solution; test & develop VoIP, SIP and IMS network; & other duties/skills as required. [Job ID: NSN-F10-TXSIE]. Send resume to: NSN Recruiter, MS 4C-1-1580, 6000 Connection Dr, Irving, TX 75039 & note specific Job ID#.

DELL MARKETING LP has an opening in Foothill Ranch, CA for Software Development Advisor to design, develop & maintain Java & JEE projects & BPM apps. Requires Bachelor's & 6 yrs experience. Send resumes to PSC-PRMResume@ps.net. Job code 11000ZV1 must be referenced in email subject line. EOE.

SYSTEMS ANALYST - Analyze, dsgn, dvlp, test & implmt s/ware components & applics utilizing MS Visual Studio 2008/2010, C#, ASP.NET, SQL Server, LINQ, WCF/Web Services, Windows Services, SOAP, .Net Framework 3.0/3.5/4.0. JavaScript, IIS 7; Frequent travel reqd. Reqs MS Comp Sci, Eng or rel. Mail resumes to Sysnet Technology Solutions, 50 Cragwood Rd., Ste. # 219, South Plainfield, NJ 07080.

Composite Software, Inc.
has the following job opportunity:

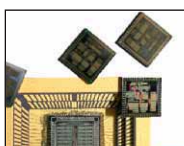
Lead Solutions Engineer

San Mateo, CA
Ref: LSE-CA

Responsible for providing development - level assets to the company's technical support team. Provide technical solutions in the area of database software and handle customer or product issues.

Mail resume to:

Composite Software, Inc.
Attn: L. Dominguez
2655 Campus Drive, Suite 200
San Mateo, CA 94403
Must reference job code LSE-CA.



Technical University of Denmark DTU

PROFESSOR
Software Engineering

DTU Informatics. Applications are invited for a professorship in Software Engineering. Applicants should demonstrate the ability to combine theoretical knowledge with development of computational solutions within one of the following (or closely related) areas: Software Engineering for mobile and distributed computing (e.g., Internet-of-Things) and Software Engineering for high performance computing.

Application deadline: **15 September 2011**

DTU is a leading technical university rooted in Denmark but international in scope and standard. Our total staff of 5,000 is dedicated to create value and to promote welfare for the benefit of society through science and technology; and our 7,000 students are being trained to address the technological challenges of the future. While safeguarding academic freedom and scientific independence we collaborate with business, industry, government, public agencies as well as other universities around the world.

Further details: dtu.dk/career

HP Enterprise Services, LLC is accepting resumes for the following positions:

Information Testing (Specialist-Senior)

Palo Alto, CA • (Ref. #RESPALIT11)

Design, develop and execute all testing-related activities on applications, infrastructure or hardware components of IT solutions, which include both third party software and internally developed applications and infrastructure. Extensive travel required to various unanticipated locations throughout the U.S.

Technology Consultant

Mountain View, CA • (Ref. #ESMVTC11)

Provide technology consulting to customers and internal project teams. Provide technical support and/or leadership in creation and delivery of technology solutions designed to meet customers' business needs and, consequently, for understanding customers' businesses.

Services Information Developer (SAP)

Beaverton, OR • (Ref. #ESBEAKCH1)

Conceptualize, design, develop, unit-test, configure, or implement portions of new or enhanced (upgrades or conversions) business and technical software solutions through SAP ERP application of appropriate standard software development life cycle methodologies and processes.

Services Information Developer

Salem, OR • (Ref. #ESSALSID11)

Conceptualize, design, develop, unit-test, configure, or implement portions of new or enhanced (upgrades or conversions) business and technical software solutions through application of appropriate standard software development life cycle methodologies and processes.

Data Engineer

Pontiac, MI • (Ref. #ESPONDE21) and Roseville, CA • (Ref. #ESROSDE11)

Conceptualize, design, develop, integrate, deploy and maintain the data layer and data components by applying specialized knowledge of information, data and database management disciplines to provide world-class solutions.

Mail resume to HP Enterprise Services, LLC, 5400 Legacy Drive, MS H1-6F-61, Plano, TX 75024. Resume must include Ref. #, full name, email address & mailing address. No phone calls please. Must be legally authorized to work in the U.S. without sponsorship. EOE.

Apple is looking for qualified individuals for following 40/hr/wk positions. To apply, mail your resume to 1 Infinite Loop 84-REL, Attn: AEC Staffing-LJ, Cupertino, CA 95014 with Req # and copy of ad. Job site & interview, Cupertino, CA. Principals only. EOE.

DFT Engineer [Req# 8185611]

Responsible for developing DFT architecture and test specifications. Requires Master's degree, or foreign equivalent, in Computer Engineering, or related and 5 years professional experience in job offered or in a related occupation, including TetraMAX and or Fastscan, Scan Compression Technologies (DFTMAX or TestKompress), BSDL, JTAG, 1500, CoreTest, Verilog, Perl.

ASIC Design Eng 3 [Req#8290469]

Work on computer architecture, specifically in areas of computer cache design, multiprocessor coherency, performance analysis and hardware verification. Req.'s Master's degree, or foreign equivalent, in Computer Science, Computer Engineering, Electrical Engineering, or related. Also professional experience with: modern digital designs; CPU pipelines; advanced microarchitecture concepts including out-of-order issue, multiprocessing, cache hierarchy & coherency protocols, and memory consistency, synchronization, and barriers; RTL code in verilog language; C and assembly programming.

Software Engineer [Req#8304642]

Design, architect, develop and maintain high performance systems. Req.'s Bachelor's degree, or foreign equivalent, in Computer Science, Information Technology, Electrical Engineering, Electronic Engineering or related plus Five (5) years professional experience in job offered or in a related occupation Professional experience must be post-baccalaureate and progressive in nature. Also experience with: Must have professional experience with: scalable server/client-server Applications; Java/J2EE/Oracle/SQL/JDBC/ in an Unix/Linux environment; developing high performance server applications using Spring, Tomcat/Netty/Jboss; performance tuning, garbage collection, memory profiling; various data-structures and their application for high performance data access and manipulation; High Availability applications and replication mechanisms.

Software Engineer [Req. #8304221]

Develop Safari Web Browser and underlying WebKit engine on desktop and embedded operating systems, using C++ programming language. Req.'s Master's degree, or foreign equivalent, in Computer Science, Electrical Engineering, Mathematics, or related plus Four (4) years professional experience in job offered or in a related occupation. Also experience with Mac OS X Software Development; Windows Software Development; Web Technology; Internationalization/International Text processing; network programming; HTTP Networking; subversion version control system; bugzilla bug tracker; XML; SQL; and C++.

Software Engineer [Ref#7520041]

Work as part of iPhone WiFi Software team responsible for designing and developing mobile wireless LAN technology for Apple products. Requires Bachelor's degree, or foreign equivalent, in Systems Engineering, Electrical Engineering, or related and 5 years professional experience in job offered or in a related occupation Professional experience must be post-baccalaureate and progressive in nature. Also experience with C; TCP/IP suite of protocols; 802.11/WiFi protocols, technologies and standards; software development experience on Mac, Windows or Linux environment.

Apple is looking for qualified individuals for following 40/hr/wk positions. To apply, mail your resume to 1 Infinite Loop 84-REL, Attn: AEC Staffing-LJ, Cupertino, CA 95014 with Req # and copy of ad. Job site & interview, Cupertino, CA. Principals only. EOE.

Point of Sale Lead Test Engineer. [Ref# 7599239]

Conduct integration testing for various Point of Sale projects. Requires Bachelor's degree, or foreign equivalent, in Engineering, Civil Engineering, or related and 5 years professional experience in job offered or in a related occupation Professional experience must be post-baccalaureate and progressive in nature. Also experience with quality assurance; industry standard processes, protocols, tools, and methodologies; writing clear and concise test plans; Unix; MySQL; hands-on scripting for test automation tools.

Mechanical Quality Engineer [Ref# 8304779]

Responsible for quality control and manufacturing concepts to develop specific Product Quality Plan (PQP) appropriate to program and commodity. Req.'s Master's degree, or foreign equivalent, in Industrial Engineering, Mechanical Engineering, or related plus 5 (Five) years professional experience in job offered or in a related occupation. Must also have professional experience with: process quality plans, supplier quality audits; drive corrective actions and failure analysis efforts; mechanical quality and reliability concepts, supplier management, and general manufacturing operations; project management experience; DFx and DOE principles. May require 35% of international travel time.

Radio System Integrator [Ref# 7519837]

Evaluate future cellular chipsets and technologies and drive their System Architecture, proof-of-concept, planning, and execution into the iPhone/iPad platform. Req.'s Bachelor's degree, or foreign equivalent, in Electrical Engineering, Computer Engineering, or related. Five (5) years professional experience in job offered or in a related occupation. Professional experience must be post-baccalaureate and progressive in nature. Also experience with: computer architecture, advanced logic/digital design; hardware design of high-volume Smartphones; wireless communications protocols such as GSM, CDMA2000, WCDMA, GPS, and WLAN; experience with integration and testing of RF components and circuits; design and validation of power supplies such as linear regulators and switched-mode regulators, DC/DC converters; design power management of radio on iPhone/iPad. Less than 20% international travel required.

ASIC Design Engineer [Ref# 8276488]

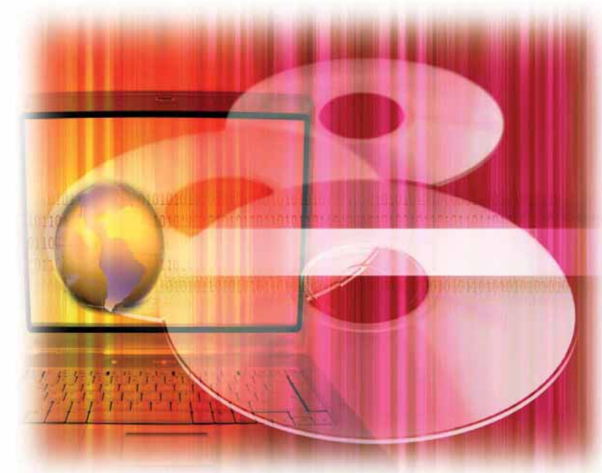
Support and develop the Virtuoso Custom Schematic and Layout Entry environment. Requires Master's degree, or foreign equivalent, in Electrical Engineering, or related and 3 years professional experience in job offered or in a related occupation, including Skill, Python and Perl programming; Spectre control; and Component Description Format language.

Software Engineer - SOA [Ref# 8398470]

Develop detailed design and deliver a scalable implementation. Req.'s Bachelor's degree, or foreign equivalent, in Computer Science, or related plus Five (5) years professional experience in job offered or in a related occupation Professional experience must be post-baccalaureate and progressive in nature. Also experience with: Java, J2EE (EJBs, servlets, JSP, struts) and XML technologies; publishing and consuming WebServices using SOA Infrastructure (ESB) and/or BPEL; enterprise messaging using EAI and JMS; designing schemas, writing DML queries and query performance analysis for relational databases such as Oracle or MySQL; Oracle Streams and Advanced Queues; grid based caching solutions like Coherence; UNIX, shell scripts and Perl; Mac OS X; 5 years hands-on programming and at least 2 more years technical leadership role.

Stereoscopic Technologies and Effects

John Bryant and Jason Bay, *Griptonite Games*



Viewing technologies available for home use offer consumers the option of creating their own stereoscopic content.

Stereoscopic effects are achieved by submitting two images to the viewer, one to each eye, depicting the same scene from slightly offset vantage points. If the two images are similar, the brain merges them into a single image with the illusion of depth.

In the past decade, there has been a boom in technology for displaying stereoscopic imagery, including parallax barriers, shutter glasses, and the polarized glasses used in movie theaters. Stereoscopic cameras have become more prevalent as well. Now, viewing technologies available for home use offer consumers the option of creating their own stereoscopic content.

STEREOSCOPIC TECHNOLOGIES FOR HOME

Many technologies exist for viewing stereoscopic content at home, ranging from the low-tech red/cyan glasses shown in Figure 1a to the precision parallax barriers and shutter glasses shown in Figure 1b. Each technology has advantages and drawbacks.

Red/cyan glasses, are an inexpensive option for viewing stereoscopic content at home because they only require a red filter over one eye and a cyan filter over the other. Any TV, monitor, or print graphic can display a red/cyan stereoscopic frame by depicting one eye's content with red and the other with cyan, allowing the glasses to filter out the left- and right-eye images based on color.

The main advantage of this option is cost, since a pair of glasses sells for around $2.00. The drawback is in color reproduction, because it's difficult to get a wide color range using this technology. Another challenge is that the TV or monitor must be correctly color calibrated to the filters on the glasses; otherwise, as Figure 2 shows, the filters won't block the left- and right-eye images effectively, and image ghosting will occur.

Parallax barriers, a technology commonly used in 3D cameras and video recorders, cover the device's view screen and orient the light from the left-eye frame to the viewer's left

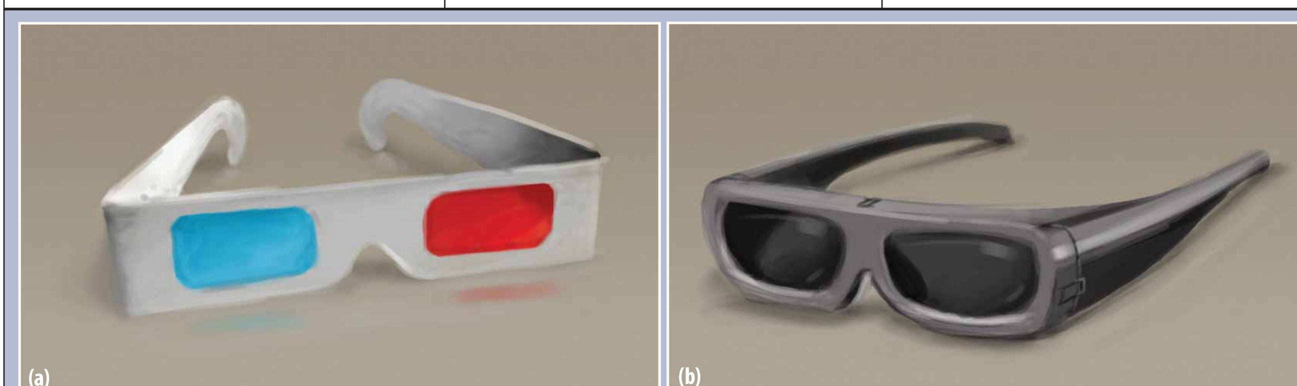


Figure 1. Technologies for viewing stereoscopic content at home include (a) red/cyan glasses and (b) shutter glasses.

Computer Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page
Qmags THE WORLD'S NEWSSTAND®


(a) (b) (c)

Figure 2. (a) An example of a stereoscopic red/cyan image and (b) left and (c) right images used for shutterglasses.

eye and from the right-eye frame to the viewer's right eye. This allows excellent color reproduction and contrast, but suffers from a narrow viewing angle because the user's eyes must be correctly aligned to the screen to perceive the 3D effect. The technology doesn't work well when used on large displays, and it halves the resolution since it sends every other pixel to a different eye.

The home entertainment industry has currently adopted shutter glasses as the stereoscopic medium of choice. This technology is based on glasses that consist of two liquid crystal screens that, when activated, can block all light from entering one eye or the other. The glasses are synchronized to a stereoscopic-capable TV or a computer monitor capable of showing the left-eye frame and then the right-eye frame alternately in quick succession, usually at 120 frames per second, while the glasses block an image from the right eye and then from the left eye every other frame. This alternation is fast enough that the brain perceives the content as one continuous image with stereoscopic depth.

Shutter glasses support a wide viewing angle and can run at a full 1,080-pixel resolution, but their main advantage is in color reproduction: they allow for any range of colors that an image or movie can display. The main drawback is a contrast problem caused by always blocking one eye or the other at any given time. Cost can also be a factor since a 3D-capable TV or monitor is required, and the glasses average around $150 per pair.

CREATING STEREOSCOPIC CONTENT

A variety of commercial software is available for creating stereoscopic pictures and movies. If a 3D video camera is available, software such as Adobe After Effects and Adobe Vegas can be used to add stereoscopic special effects and text. Maya or 3D Studio Max are good options for generating content from 3D computer graphics as both can be configured to output stereoscopic content.

Freeware and shareware packages are also an option. StereoPhoto Maker allows the creation of stereoscopic stills and also has slideshow capabilities, whereas StereoMovie Maker supports the creation of stereoscopic video and supports a variety of formats. To incorporate stereoscopic content into webpages, there are several Java-based applets available such as Stereoscope and Stereoscopic Applet.

Many video cards enable the creation of immersive stereoscopic applications and video games using graphics APIs for DirectX or OpenGL. Each API includes instructions and considerations for creating stereoscopic-capable programs. Most new games take advantage of this to offer support for stereoscopic effects using shutter glasses technology, including *World of Warcraft, Starcraft*, and *Crysis 2*.

ACHIEVING A STRONG STEREOSCOPIC EFFECT

When creating stereoscopic content there are a few guidelines that should be followed to achieve great-looking pictures and videos without causing undue eye strain for the viewer.

One technique content creators use is to make objects appear to "pop out" of the screen and literally appear to float in front of the viewing device. This emphasizes the stereoscopic effect but can be problematic if handled incorrectly.

When creating a pop-out effect, it's necessary to make sure the object doesn't touch any edge of the screen. This is important because the brain must make assumptions about the left- and right-eye images it's given; if presented with an object that appears to be in front of the screen but at the same time is obstructed by the screen edge, the visual contradiction can destroy the stereoscopic effect. This problem will occur if scene objects touch the top, bottom, or side edges.

Due to the offset nature of stereoscopic images, it's important for objects to avoid touching the screen edges when using a pop-out effect. The effect requires a shift to the left and right of "center" for the object, and when an object is up against an edge there's no room to shift the image, resulting in a reduction in the stereoscopic effect. This particular problem only occurs with side edges since there's no vertical shifting required to achieve stereoscopy.

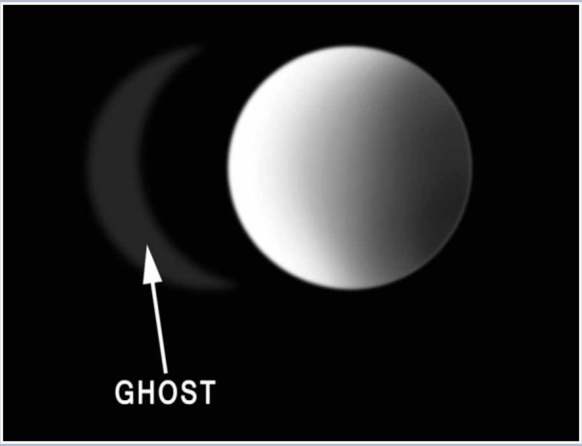


Figure 3. Image ghosting usually appears as a faint offset image or "halo" of a high-contrast object in a scene.

Effects that appear to be positioned beyond or behind the viewing medium aren't impacted as much by touching the edges of the screen because this doesn't create a visual contradiction. However, very large objects can still lose some of the depth effect if they can't be shifted enough to the left or right. Therefore, content creators should try to keep large objects centered as much as possible.

It's important to avoid using extreme pop-out or depth for long periods of time. One reason is that such effects require more horizontal shifting, and it can be difficult to center an object enough to allow for the full shift necessary. Another reason is that it can be hard for the brain to reconcile extreme differences in the left and right images, leading to headaches and eye fatigue.

Image ghosting can become an issue with any stereoscopic medium because most of the technologies inadvertently allow some of the left image to enter the right eye and vice versa. As Figure 3 shows, ghosting will usually appear as faint offset images or "halos" of high-contrast objects in a scene, such as a bright white ball against a black background. The best way to avoid ghosting is to avoid high contrast between objects at different stereoscopic depths.

Developers are working on new methods that will resolve many of the problems related to stereoscopic technologies. For example, Sony is introducing a new organic light-emitting diode (OLED) glasses system that will dramatically reduce the contrast and ghosting issues. In addition, TV manufacturers are experimenting with larger sizes for parallax barriers so that glasses are no longer needed to view full-color stereoscopic content.

The future is looking bright for users being able to create and experience life-like and truly immersive 3D worlds and environments, all from the comfort of their own home. ■

John Bryant is a Technical Manager at Griptonite Games (http://www.griptonite.com). Contact him at jbryant@griptonite.com.

Jason Bay is the Studio Technical Director at Griptonite Games. Contact him at jbay@griptonite.com.

Images provided by ***Jack Scott Hill****.*

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

Knowledge Representation and Reasoning for Intelligent Software Systems

Emil Vassev and Mike Hinchey,
Lero—the Irish Software Engineering Research Centre

A successful intelligent software system employs its knowledge to become more self-aware.

The concept of intelligence is built upon four fundamental elements: data, information, knowledge, and wisdom (www.makhfi.com). In general, data takes the form of measures and representations of the world—for example, raw facts and numbers. Information is obtained from data by assigning relevant meaning, such as a specific context. Knowledge is a specific interpretation of information. And wisdom is the ability to apply relevant knowledge to a particular problem. Thus, wisdom requires awareness, judgment, rules, and eventually experience; it also helps create new knowledge.

When developing intelligent software systems, designers employ different kinds of knowledge to derive models of specific domains of interest. There's no standard classification system—the problem domain determines what kinds of knowledge designers might consider and what models they might derive from that knowledge. For example, knowledge could be internal (about the system itself) or external (about the system environment). Knowledge could also be a priori (initially given to a system) or from experience (gained from analysis of tasks performed during the system's lifetime). Other kinds of knowledge might relate to the application domain, the system's structure, problem-solving strategies, the system's ability to communicate with other systems, and so on.

KNOWLEDGE REPRESENTATION

Intelligent system designers can use different elements to represent different kinds of knowledge. Knowledge representation (KR) elements could be primitives such as rules, frames, semantic networks and concept maps, ontologies, and logic expressions. These primitives might be combined into more complex knowledge elements. Whatever elements they use, designers must structure the knowledge so that the system can effectively process and it and humans can easily perceive the results.

Rules

Rules organize knowledge into premise-conclusion pairs, in which the premise is a Boolean expression and the conclusion a series of statements. The premise is wrapped in an IF…THEN block and consists of one or more clauses, with multiple clauses connected by logical operators such as AND, OR, and NOT. For example: *IF it's lunchtime OR I'm hungry THEN I shall go to the restaurant.*

A major advantage of rule-based KR is its extreme simplicity, which makes it easy to understand the knowledge content. Rules that fire under specific conditions readily demonstrate the reasoning. However, a rule-based KR model can grow very large, incorporating thousands of rules and requiring extra effort and tools to maintain their consistency.

Frames

Frames represent physical entities, such as objects or persons, or simple concepts via a collection of information, derivation function calls, and output assignments, and can contain

Published by the IEEE Computer Society

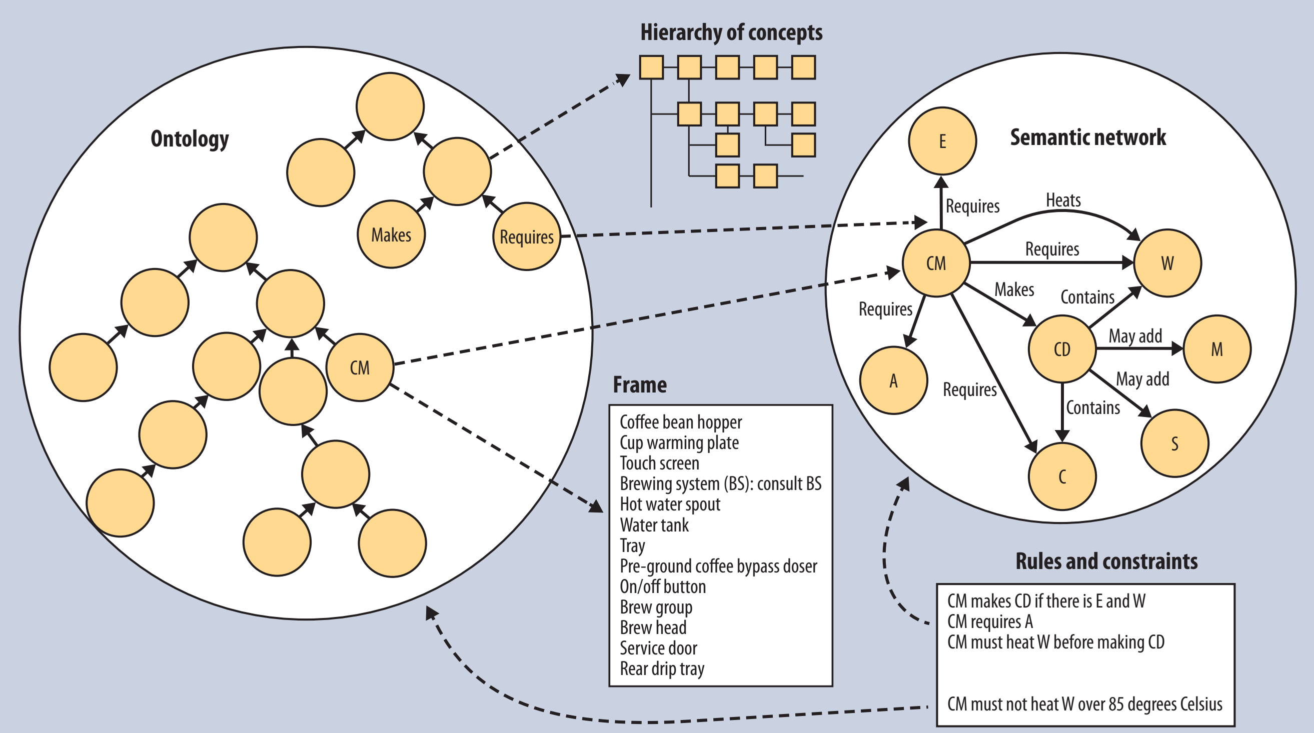


Figure 1. Ontology representing the concept of a coffee machine (CM).

descriptions of semantic attributes as well as procedural details. Frames contain two key elements: *slots* are sets of attributes of the described entity, with special daemons often included to compute slot values, and *facets* extend knowledge about an attribute.

Semantic networks and concept maps

Knowledge is often best understood as a set of related concepts. A semantic network is a directed graph consisting of nodes—which represent concepts—connected by edges—which represent semantic relations between those concepts. There's no standard set of relations between concepts in semantic networks, but the following relations are common:

- instance: *X* is an instance of *Y* if *X* is a specific example of the general concept *Y*
- isa: *X* isa *Y* if *X* is a subset of the more general concept *Y*
- haspart: *X* haspart *Y* if the concept *Y* is a part of the concept *X*

Inheritance is a key notion in semantic networks and can be represented naturally by IsA relations. Essentially, a computer-based semantic network uses metadata (data describing data) to represent the meaning of different information. Intelligent systems that recognize the meaning of information—for example, data stored in a warehouse—become immeasurably more intelligent. Extensible Markup Language (XML) and Resource Description Framework (RDF) are common content-management schemes that support semantic networks.

Concept maps are similar to semantic networks, but they label the links between nodes in very different ways. They are considered more powerful than semantic networks because they can represent fairly complex concepts—for example, a hierarchy of concepts with each node constituting a separate concept. Concept maps are useful when designers want to use an intelligent system to adopt a constructivist view of learning.

Ontologies

Ontologies inherit the basic concepts provided by rules, frames, semantic networks, and concept maps. They explicitly represent domain concepts, objects, and the relationships among those concepts and objects to form the basic structure around which knowledge can be built (W. Swartout and A. Tate, "Ontologies," *IEEE Intelligent Systems*, Jan./Feb. 1999, pp. 18-19). The main idea is to establish standard models, taxonomies, vocabularies, and domain terminology and use them to develop appropriate knowledge and reasoning models.

An ontology consists of hierarchies of concepts—for example, an "objects" concept tree or a "relations" concept tree. Each concept has properties, which can be regarded as a frame. The relationships among the concepts form semantic networks, and rules and constraints impose restrictions on the relationships or define true statements in the ontology (facts).

Figure 1 shows an ontology that represents the concept of a coffee machine. CM has properties such as height, weight, coffee bean hopper, touch screen, container, and so on. A semantic network defines the relationships between CM and the rest of the concepts in the ontology and includes the following properties: CM requires E (electricity), CM requires A (action), CM requires C (coffee), CM requires W (water), and CM makes CD (coffee drink). Some rules expressed with the ontology concepts add new knowledge about the coffee machine.

Logic

To achieve the precise semantics necessary for computational purposes, intelligent system designers often use logic to formalize KR. Moreover, logic is relevant to reasoning (inferring new knowledge from existing knowledge), which in turn is relevant to entailment and deduction (R.J. Brachman and H.J. Levesque, *Knowledge Representation and Reasoning*, Elsevier, 2004).

The most prominent logical formalism used for KR is *first-order logic*. FOL helps to

- describe a knowledge domain as consisting of objects, and
- construct logical formulas around those objects.

Similar to semantic networks, statements in natural language can be expressed with logic formulas describing facts about objects using predicate and function symbols.

Extensions of FOL such as *second-order logic* and *temporal logics* strive to improve the logic formalism by increasing expressiveness. The problem with FOL is that it can quantify over individuals, but not over properties and time—we can thus specify a property's individual components, but not an individual's properties. With SOL, for example, we can axiomatize the sentence "component A and component B have at least one property in common, such as sharing at least one interface," which we can't do with FOL. Temporal logics make it possible to model knowledge either as linear time or branching time temporal models, and can be used to describe and formalize complex reasoning patterns prescribing inference steps operating over temporal knowledge models.

Another prominent formalism is *description logic*, which evolved from semantic networks. With DL, we represent an application domain's knowledge by first defining relevant concepts in TBox and then using ABox to specify properties of objects. While less expressive than FOL, DL has a more compact syntax and better computational characteristics.

Knowledge consistency is critical for efficient reasoning.

COMPLETENESS AND CONSISTENCY

No KR model can provide a complete picture of the domain of interest. Domain objects are often real-world entities that can't be described by a finite set of symbolic structures; moreover, such objects don't exist in isolation but in unlimited contexts. Intelligent systems consequently must rely on reasoning to infer missing knowledge.

Knowledge consistency is critical for efficient reasoning. The degree to which systems achieve this efficiency is determined by whether they assume that the operational world is complete and closed or incomplete and open:

- *closed-world assumption* (CWA)—unless an atomic sentence is known to be true, it can be assumed to be false; and
- *open-world assumption* (OWA)—any information not explicitly specified, or that can't be derived from known data, is considered unknown.

The following example illustrates the difference between these two assumptions:

Given:	Emil drives a Mazda.
Question:	Does Emil drive a red Mazda?
Answer:	(CWA) No. (OWA) Unknown. (Emil's Mazda could be red.)

Note that FOL imposes CWA semantics, and DL imposes OWA semantics. Although more restrictive than OWA, CWA maintains consistency in knowledge because it doesn't allow adding new facts, which can lead to inconsistency.

Intelligent systems can also employ consistency rules and constraints, such as "no negation," to preserve knowledge consistency. There could be constraints for knowledge acquisition, retrieval, updating, and inferences.

REASONING

When an intelligent system needs to decide on a course of action and there's no explicit knowledge to guide this decision, the system must reason—that is, figure out what it needs to know from what it already knows. There are two basic types of reasoning:

- *monotonic*—new facts can only produce additional beliefs; and
- *nonmonotonic*—new facts will sometimes invalidate previous beliefs.

Current reasoning mechanisms are far from efficient, which is partially due to KR's inherently challenging task. FOL- or DL-based inferential engines usually do computations, acting on existing knowledge to produce new knowledge.

FOL-based inferential engines use automated-deduction algorithms to prove theorems and build finite models, often in parallel. Theorem proving can help find contradictions or check for new information, while finite model building is a complementary inference task. The problem with FOL-based inferences is that the logical entailment for FOL is semidecidable—that is, if the desired conclusion follows from the premises, then eventually resolution refutation will find a contradiction. As a result, queries often unavoidably don't terminate.

Inference engines based on DL are extremely powerful when reasoning about taxonomic knowledge, as they can discover hidden subsumption relationships among classes. However, their expressive power is restricted to reduce computational complexity and to guarantee the decidability of their deductive algorithms. This restriction effectively prevents the wide application of taxonomic reasoning to heterogeneous domains.

To make reasoning more efficient, intelligent systems should also include mechanisms capable of sifting context-aware knowledge from the overwhelming amount of information that's irrelevant to the current context.

A successful intelligent system employs its knowledge to become more self-aware. To achieve this self-awareness, system designers are developing more sophisticated KR models and reasoning capabilities, drawing on research in ontologies, data mining, intelligent agents, autonomic computing, knowledge processing, and many other areas.

Emil Vassev *is a research fellow at Lero—the Irish Software Engineering Research Centre, University of Limerick, Ireland. Contact him at emil.vassev@lero.ie.*

Mike Hinchey, *the Software Technologies column editor, is director of Lero and professor of software engineering at the University of Limerick. Contact him at mike.hinchey@lero.ie.*

Interactive Computing on Wheels

Manfred Tscheligi, Alexander Meschtscherjakov, and David Wilfinger,
University of Salzburg

Automotive designers are exploiting invisible computing technologies to create a better user experience within a multifaceted design space.

Since Carl Benz patented the first gasoline-powered automobile, the Motorwagen, on 29 January 1886, cars have transformed the world like no other technology. Roads cover the landscape like the veins of a body, and motor vehicles constitute the pulse of modern civilization, providing humans with almost unlimited mobility to work, play, and travel.

In recent years, cars have evolved from mere mechanical machines, designed to transport people from one place to another, to highly interactive, networked computing devices. Automotive engineering and information technology are converging and will soon deliver highly sophisticated systems that dramatically increase driver safety, comfort, and convenience.

AUTOMOTIVE DESIGN AND USER INTERACTION

Modern cars contain a multitude of computing systems. Many of these systems—including antilock breaking, traction control, and stability control systems—actively support user safety via contextual interventions. Other pervasive computing systems, such as adaptive cruise control and lane-departure warning systems, draw the driver's attention to critical situations. Still other systems make driving more comfortable (automatic climate control, electronic seat adjustment, remote radio controls) and enjoyable (entertainment systems) or provide valuable information (navigation devices).

In fact, modern vehicles are beginning to resemble a high-end computer more than a car in the traditional sense, with onboard systems supporting a growing number of secondary and tertiary tasks such as locating a nearby friend or restaurant. This is partly a response to changing cultural attitudes. People are used to being connected all the time, and they want to be able to communicate with family, friends, and colleagues without interruption even while driving. The car is no longer simply a means to get from point A to point B—it's also a medium for socializing, business, and entertainment.

The demand to customize vehicles to fit a driver's personal needs is causing a paradigm shift in automotive design from closed systems, for which auto manufacturers exclusively dictated the design, toward open standards and application platforms. Vehicle-to-vehicle (V2V), road-to-vehicle (R2V), and vehicle-to-infrastructure (V2I) communication technologies are enabling a wide array of new interaction possibilities. The seamless integration of nomadic devices and cloud-based personal information access will be a big challenge for automotive user interface research in the next decade.

New interactive capabilities present another challenge in that they increase driver distraction and thus decrease safety. As long as cars must be manually directed—fully automated steering remains in the distant future—the driver must focus on the driving task, with eyes on the road and hands on the wheel. The integration of automotive, Internet, and mobile technologies will thus require novel design approaches that consider these technologies' impact on driving behavior.

Published by the IEEE Computer Society

Computer | Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page

INNOVATIVE AUTOMOTIVE USER INTERFACES

For automotive UI designers, the question is how to make the computer disappear (D.A. Norman, *The Invisible Computer,* MIT Press, 1999) and thus become what Mark Weiser called a "calm technology" ("The Computer for the 21st Century," *Scientific American,* Sept. 1991, pp. 94-104). That is, how do we move from today's overloaded cockpits to the minimalistic, aesthetically pleasing UIs of many concept cars? Can we provide the same functionality while simultaneously delivering a better user experience with a reduced cognitive workload and fewer distractions?

The current state of the art in automotive UI technology includes speech-based interaction, touch screens, central control units with center-stack displays, steering-wheel buttons, and head-up displays. Thus far, however, none of these input and output modalities have proven completely effective—either they all must be synergized or new ones must be envisioned.

Persuasive interfaces

Academic and industrial researchers are exploring the use of interfaces that integrate with ambient technologies and provide contextual information designed to foster safer and more fuel-efficient driving behavior. These interfaces rely on persuasive strategies such as reducing complex behaviors to simple tasks, self-monitoring, and social comparison (B.J. Fogg, *Persuasive Technology: Using Computers to Change What We Think and Do,* Morgan Kaufmann, 2003).

For example, Figure 1 shows a proposed persuasive in-car UI that uses a set of green leaves to visualize fuel efficiency. The better a driver performs, the more green leaves grow; when driving is wasteful, leaves vanish. At the end of each trip, the system calculates an EcoScore and compares it to that of other drivers. To facilitate eco driving in hybrids, various auto manufacturers have developed similar interfaces, including Ford's SmartGauge with EcoGuide dashboard and Honda's ambient meter.

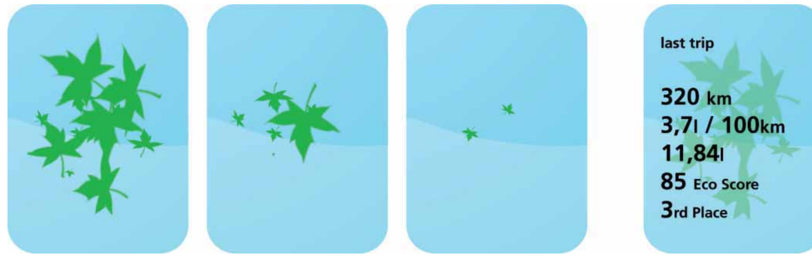


Figure 1. Design sketch for a persuasive interface to foster fuel-efficient driving behavior. The interface uses a set of green leaves to visualize fuel efficiency. The better a driver performs, the more green leaves grow; when driving is wasteful, leaves vanish. At the end of each trip, the system calculates an EcoScore and compares it to that of other drivers.

Novel input/output modalities

Novel driver interfaces will also need to address limitations to driver visibility and accessibility. Modern cars either split up input and output modalities (for example, BMW's iDrive, Mercedes' COMAND, and Audi's Multi Media Interface), utilize touch screens, rely on speech input and output, or use a hybrid solution to allow interaction with the different in-car services. New approaches could move touch screens from the center stack onto the steering wheel or combine haptic input modalities on the steering wheel (T. Döring, "Gestural Interaction on the Steering Wheel: Reducing the Visual Demand," *Proc. Ann. Conf. Human Factors in Computing Systems* (CHI 11), ACM Press, 2011, pp. 483-492).

Figure 2 illustrates an intelligent steering wheel (ISW) developed by AUDIO MOBIL Elektronik GmbH. This prototype ISW is equipped with an integrated touch screen that visualizes instrument panel, infotainment, and navigation information. All cockpit controls—turn indicator, light switch, and so on—are fully integrated into the steering wheel and can be activated via either the touch screen or buttons at the top of the wheel. The ISW ensures short distances for both input and output.

© AUDIO MOBIL 2009

Figure 2. **Intelligent steering wheel (ISW) developed by AUDIO MOBIL Elektronik GmbH. This prototype is equipped with an integrated touch screen that visualizes instrument panel, infotainment, and navigation information. All cockpit controls—turn indicator, light switch, and so on—are fully integrated into the steering wheel and can be activated via either the touch screen or buttons at the top of the wheel. The ISW ensures short distances for both input and output.**

Figure 3 shows an alternative steering-wheel UI we are developing

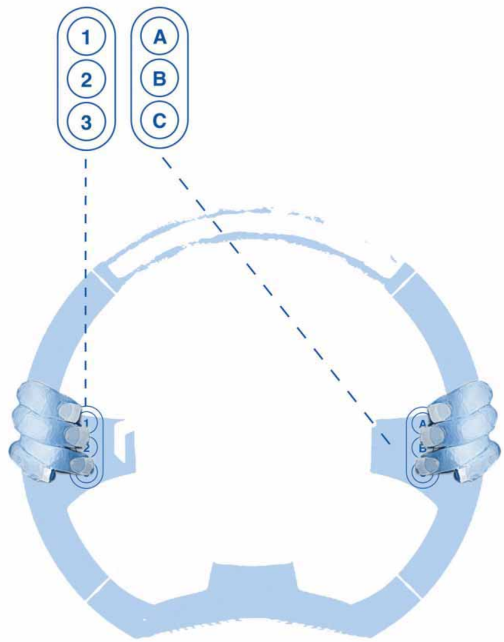


Figure 3. Back-of-steering-wheel UI concept. Touch-sensitive strips situated on the back of the steering wheel, similar to foot pedals, are divided into three zones that can function as sliding elements or as physical buttons with tactile feedback and potentially a clicking sound. In combination with a head-up display, the system lets drivers keep their hands on the wheel and eyes on the road.

at the University of Salzburg's Christian Doppler Laboratory on Contextual Interfaces. Like other devices such as mobile phones, the back of a steering wheel offers an easily accessible space for input modalities. Touch-sensitive strips situated on either side of the back of the steering wheel, like foot pedals, are divided into three zones that can function as sliding elements or as physical buttons with tactile feedback and potentially a clicking sound. In combination with a head-up display, the system lets drivers keep their hands on the wheel and eyes on the road.

USER EXPERIENCE MODELS

Given the diverse interaction possibilities that ubiquitous computing offers, researchers must develop new user experience models to facilitate the creation of novel UIs that will be trusted and accepted. These models must consider pragmatic factors—distraction from the driving task, cognitive workload, stress, perceived safety, and control—as well as emotive elements such as aesthetics, fun and enjoyment, and social interaction both within the vehicle and with the outside world.

Thus far, efforts to assess user experiences in the automotive context have relied on lab-based driving simulators. However, these studies have the disadvantage of considering only short-term interactions in an artificial environment. Because driving a car is highly context-dependent (for example, weather, type of street, amount of traffic), researchers must employ more in situ methods to perform a holistic evaluation of the user experience. This includes factoring in-car GPS data, controller area network (CAN) bus information, long-term usage data, and feedback methodologies like crowdsourcing.

In addition, future in-car UIs must adopt a wider social context that includes the passengers as well as the driver. Researchers must look beyond driving-related tasks to consider social activity within the vehicle and how this could influence the driver.

We suggest dividing the vehicle interior into three different design spaces: one for the driver, one for the front-seat passenger, and one for the rear-seat passengers. UI designers should consider how to maximize human-machine and social interaction potential within each individual space as well as synergistically, ideally in a manufacturer-independent way. For example, could a front-seat passenger perform certain tasks to reduce the driver's workload—effectively becoming a co-driver? Can designers make it easier for different passengers to listen to different types of music?

Tomorrow's intelligent car will be more than just a transportation mechanism—it will be a node in a complex ambient space. Invisible computing technologies have enormous potential to frame this space and shape future driving and mobility habits. ■

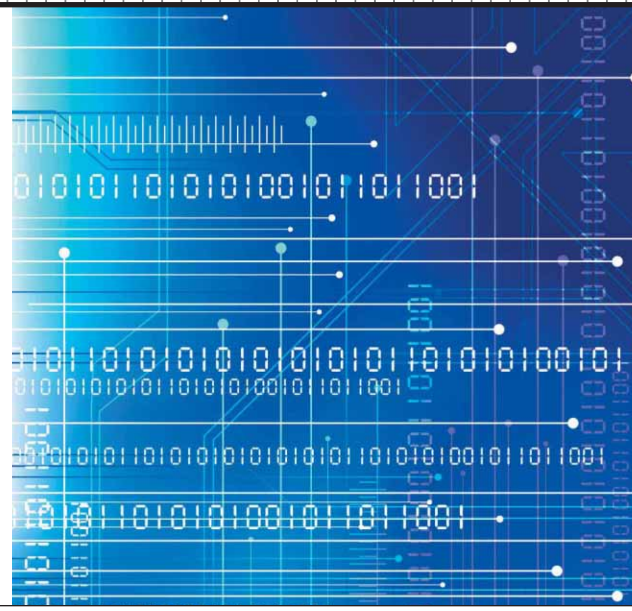
Manfred Tscheligi *heads the Christian Doppler Laboratory on Contextual Interfaces at the ICT&S Center of the University of Salzburg, Austria. Contact him at manfred.tscheligi@sbg.ac.at.*

Alexander Meschtscherjakov *is a human-computer interaction researcher in the Christian Doppler Laboratory on Contextual Interfaces at the University of Salzburg. Contact him at alexander.meschtscherjakov@sbg.ac.at.*

David Wilfinger *is a human-computer interaction researcher in the Christian Doppler Laboratory on Contextual Interfaces at the University of Salzburg. Contact him at david.wilfinger@sbg.ac.at.*

Editor: Albrecht Schmidt, University of Stuttgart, Germany; albrecht@computer.org

Using Data Mining to Help Design Sustainable Products



Manish Marwah, Amip Shah, Cullen Bash, and Chandrakant Patel, *HP Labs*

Naren Ramakrishnan, *Virginia Tech*

Data mining techniques make it possible to automate product life-cycle assessment with reasonable accuracy, even in cases of low-quality inventory data.

When making purchases, customers increasingly consider products' environmental impact as well as traditional criteria such as cost and features. In 2008, market research firm Gartner estimated that about 75 percent of enterprises would include some type of life-cycle environmental assessment in their purchasing decisions about future IT systems (D. Plummer et al., "Gartner's Top Predictions for IT Organizations and Users, 2008 and Beyond," report G00154035, 8 Jan. 2008). The recent surge in ecolabels and green stickers also indicates growing public sentiment that consumer offerings should meet minimum sustainability requirements or limit harm to the environment.

A product's environmental footprint is typically estimated through *life-cycle assessment* (LCA), which takes a comprehensive view of multiple environmental impacts such as greenhouse gas emissions, toxicity, and carcinogenicity (A. Shah et al., "Assessing ICT's Environmental Impact," *Computer*, July 2009, pp. 91-93). Researchers could use LCA to answer questions such as: How do Apple iPad, Samsung Galaxy Tab, or HP TouchPad compare in terms of their carbon footprint? Is an e-reader more environmentally friendly than a paper book? (D. Goleman and G. Norris, "How Green Is My iPad?" *The New York Times*, 4 Apr. 2010).

However, LCA can be a manual and laborious process. Accurately estimating the environmental impact factors associated with a server, for example, may involve creating a detailed inventory of all its components, usually down to parts such as integrated circuits (ICs), resistors, fans, heat sinks, and even screws, paint, and labels; estimating their mass or volume; and, finally, mapping each component to representative entries in an environmental database.

We treat LCA as a data mining problem and propose an automated LCA (auto-LCA) approach that lets a user simply input on existing product inventory and obtain an approximate environmental footprint of all its components as output.

AUTO-LCA

Given the large-scale manual processing of semistructured data associated with performing LCA, we redefine LCA as a data mining problem and integrate data mining solutions from different contexts to obtain an auto-LCA methodology.

We consider a product inventory, such as a bill of materials (BOM), to be a compositional containment hierarchy and represent it as a tree. For example, a desktop computer contains a printed circuit board (PCB), which contains ICs, capacitors, and resistors; these components in turn contain silicon and other metals.

An auto-LCA system requires two databases: a product database, which includes each product's BOM as well as information such as part number and description; and an environmental database of generic information about the environmental impact of various components—for example, the Ecoinvent database (www.ecoinvent.ch).

As Figure 1 shows, the path from BOM to environmental footprint

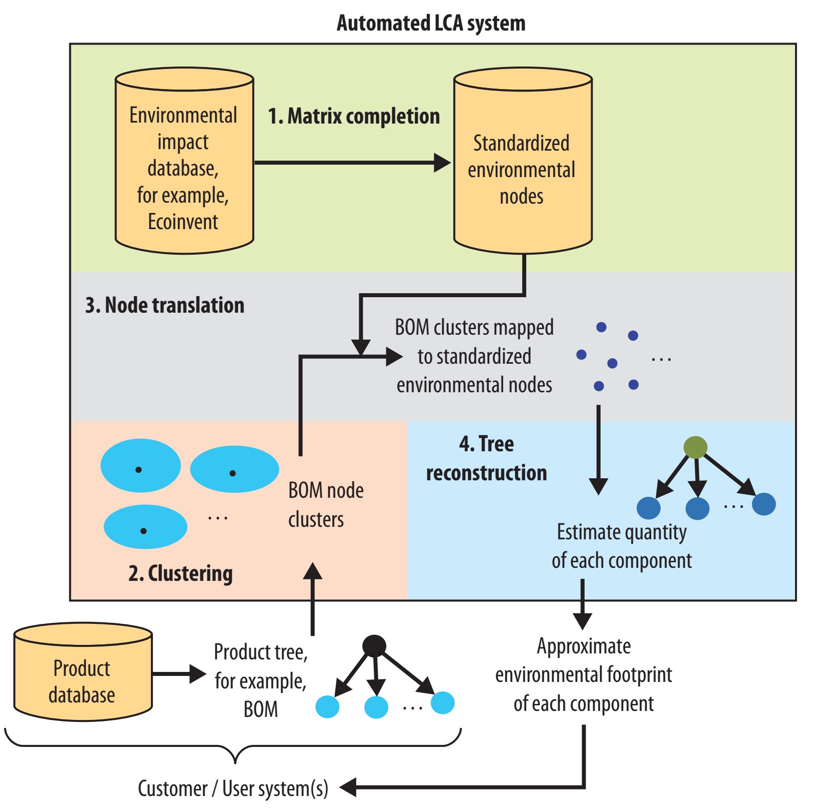


Figure 1. Automated life-cycle analysis (auto-LCA) methodology. Obtaining a product's approximate environmental footprint from its bill of materials (BOM) entails four main steps: matrix completion, clustering, node translation, and tree reconstruction.

entails four main steps: matrix completion, clustering, node translation, and tree reconstruction.

Matrix completion

We first check the environmental database for incomplete or invalid information. For example, many such databases only have a few impact factors, such as energy and ecotoxicity, listed for certain nodes, and lack relevant information such as carbon footprint data or quantified impacts on human health. The environmental database can be viewed as a matrix with components as rows and impact factors as columns. Estimating missing values thus constitutes a matrix completion task. This problem is similar to that encountered in recommender systems, in which the goal is to estimate missing ratings for unseen items such as movies and books.

Clustering

Next, we perform cluster analysis on the potentially hundreds of items listed in the BOM with the objective of grouping similar nodes—those likely to have comparable environmental impact—together. The clustering algorithm requires a distance metric computed from the node attributes to posit groupings. Simply using part numbers isn't sufficient, as many BOM components could be quite similar from the standpoint of environmental impact but very different in terms of how they're identified in the product tree. For example, two identical stainless steel screws that reside in different parts of the system might have distinct part numbers.

To compute the distance between node descriptions, we use approximate string-matching techniques such as longest common subsequence (LCS), longest common prefix (LCP), Levenshtein distance (LD), or a combination of these. Once we obtain a distance metric, we employ clustering algorithms such as *k*-medoids to group similar BOM nodes together. The resulting clusters reduce the number of parts to be evaluated from up to several thousand to a smaller, more manageable number—typically, at least an order of magnitude lower.

Node translation

We then assign each of these clusters a representative node similar to its medoid from the environmental database. In this way, we "translate" BOMs associated with distinct products that come from various suppliers and have different naming schemes into a standard terminology derived from the environmental database, thereby yielding insight into the environmental impact related to each cluster. Ideally, such translation would be learned based on some training data or by comparing BOM and environmental node descriptions, but we currently perform this translation manually. It's worth noting that clustering allows such manual translation, as it reduces the number of required translations by more than an order of magnitude.

Tree reconstruction

A challenge for translation is that the units specified in the BOM nodes and the environmental database nodes can differ. For example, most product BOMs specify the number of repeating instances for a particular part, while the environmental nodes could be specified by mass (kg). To rectify this, we use the property that for any environmental impact, the sum of the child node impact values approximately equals that of the parent (root), forming a linear system of simultaneous equations with the coefficients of the child nodes being the unknowns. To fully reconstruct the BOM tree comprising environmental nodes, we must estimate these coefficients.

We perform a least squares regression fit to best estimate the coefficients. Because the coefficients must be positive, the goal is to obtain a non-negative least squares (NNLS) fit (C.L. Lawson and R.J. Hanson, *Solving Least Squares Problems*, Society for Industrial and Applied Mathematics, 1995). Knowing a single node's weight (the root or one of the child nodes) lets us compute the environmental contribution of each child node to the total (parent) impact.

CASE STUDY: SERVER PCB

With these building blocks in place, it's possible to estimate the environmental footprint of an arbitrary product tree or BOM. We illustrate the approach by analyzing a real PCB from an enterprise server. This PCB BOM contains about 560 components, including a mix of resistors, capacitors, ASICs, and logic devices.

We cluster the BOM nodes using *k*-medoids to identify 22 unique clusters. Figure 2 shows two examples. It's relatively easy to translate these clusters (actually their medoids) into a list of nodes from the environmental database. For the resulting environmental tree, we can utilize the impact factors available from the environmental database and successfully solve for the coefficients of the child nodes. Figure 2 includes the resulting coefficients of select nodes in the environmental tree, which enables readily computing each child node's environmental impact. The median error between the sum of the child node impact factors and the parent, for about 200 impact factors, is only 12.8 percent, a highly satisfactory result.

After obtaining each child node's environmental footprint, we perform an environmental "hotspot" analysis. This essentially involves generating a Pareto list of the largest environmental contributors to the overall PCB footprint so that a designer or LCA practitioner can focus on those areas requiring further effort.

As Figure 3 shows, due to their upstream manufacturing, ICs are the biggest contributor to the PCB's overall carbon footprint, followed by the use of copper in the connectors and capacitors. These automated results match those obtained from a manual LCA, and are easy to under-

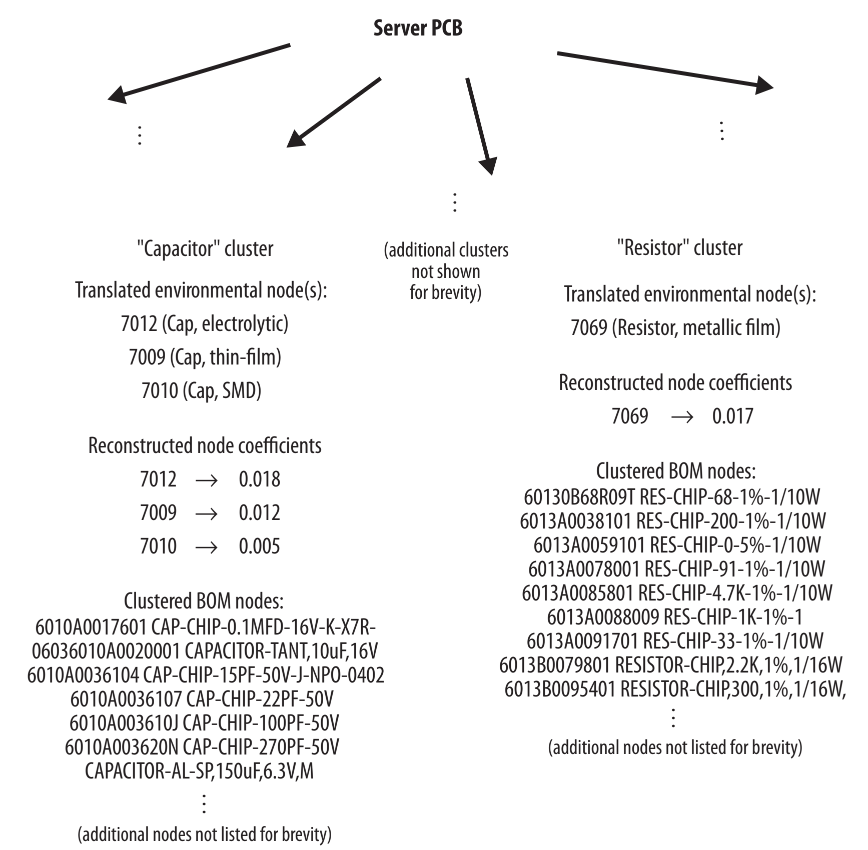


Figure 2. Server printed circuit board (PCB) clusters, node translations, and coefficients.

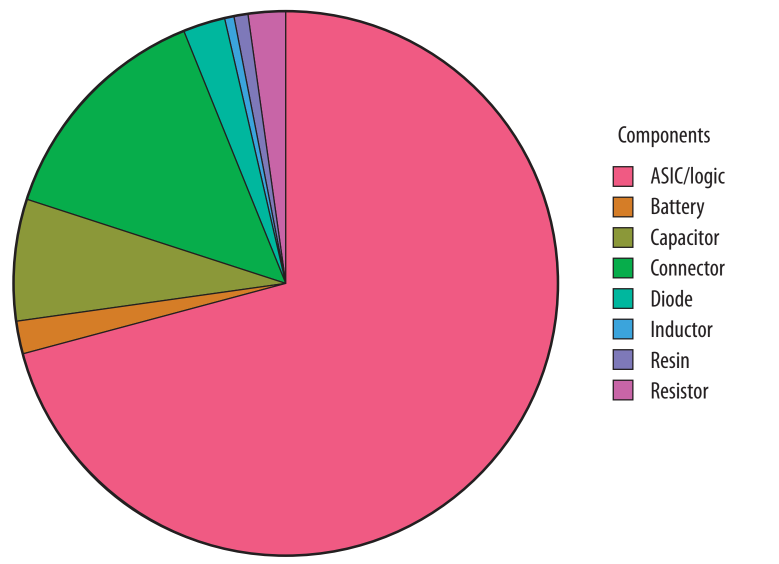


Figure 3. Results of environmental "hotspot" analysis. ICs are the biggest contributor to the PCB's overall carbon footprint.

stand even for those unfamiliar with LCA or environmental impact analysis.

PCB designers can use our auto-LCA approach to assess their design's sustainability against that of other designs as well. We anticipate eventually creating a tool that automatically scans similar ICs preloaded into the environmental database to aid in this assessment.

Consumers as well as enterprises today demand more information about products' environmental impact. Data mining techniques such as matrix completion, clustering, node translation, and tree reconstruction make it possible to automate LCA with reasonable accuracy and within fairly broad constraints, even in cases of low-quality inventory data. In the future, we plan to further evaluate these algorithms' scalability and test auto-LCA on a wider variety of systems.

Manish Marwah *is a senior research scientist at HP Labs, Palo Alto, California. Contact him at manish.marwah@hp.com.*

Amip Shah *is a senior research scientist at HP Labs, Palo Alto, California. Contact him at amip.shah@hp.com.*

Cullen Bash *is a distinguished technologist at HP Labs, Palo Alto, California. Contact him at cullen.bash@hp.com.*

Chandrakant Patel *is a senior fellow at HP Labs, Palo Alto, California. Contact him at chandrakant.patel@hp.com.*

Naren Ramakrishnan, *the Discovery Analytics column editor, is a professor of computer science at Virginia Tech. Contact him at naren@cs.vt.edu.*

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

Toward Developing an Education App Store

Charles Severance, *University of Michigan*

Creating a friction-free marketplace for the exchange of software and content for teaching and learning would contribute to solving the problem of how to best use this technology.

Since the Web first emerged in 1994, many of us in the field of teaching have been struggling with learning how to best use this technology.

The first explorers into the new frontier were naturally computer scientists interested in teaching and learning (R.J. Vetter and C. Severance, "Web-Based Education Experiences," *Computer*, Nov. 1997, pp. 139-141), resulting in an extended period of many independent experiments. Sometimes, a small project with a few faculty members would develop enough functionality to gain a small on-campus following, rising to the level where it would be further deployed.

After reaching a critical mass, early successful efforts to develop educational technology would often be transferred to a start-up company. For example, in 1996, WebCT was formed based on work from the University of British Columbia (http://en.wikipedia.org/wiki/History_of_virtual_learning_environments). In 1999, Blackboard was based on research at Cornell University, and, in the same year, Desire2Learn came out of the University of Waterloo. Angel Learning was launched in 2000 based on work at Indiana University. Developed at George Washington University starting in 1997, Prometheus was initially distributed through a consortium model to other universities, but was then sold to Blackboard (http://citl.gwu.edu/pages/projectprometheus.html), which later would acquire WebCT and Angel to consolidate market share.

VIRTUAL LEARNING ENVIRONMENTS

Starting in 2001, a series of open source projects such as CourseWork, Moodle, LON-CAPA, and OLAT were developed and remained open source. In 2004, Stanford University (CourseWork), Indiana University (OnCourse), the University of Michigan (CHEF), and MIT (Stellar), which had each developed its own virtual learning environment, formed the Sakai Project to collectively develop a single VLE they all could use. The Andrew W. Mellon Foundation provided a $2.3 million grant to help defray the costs for the four schools to transition to the Sakai software. The William and Flora Hewlett Foundation provided additional funding to develop a partners program to promote the software and recruit partners and adopters beyond the four founding schools.

The simple logic of developing a collectively produced, open source learning-management system resulted in a rapid expansion of the Sakai Partners Program, which included 120 members by the end of 2005. The University of California, Berkeley; Cambridge University; the University of Cape Town; Lancaster University, and many other schools from around the world joined the Sakai effort.

By 2010, the VLE market had reached a level of maturity and stability. Blackboard and Desire2Learn were solid commercial solutions, and Sakai and Moodle were popular open source offerings. In many ways, it felt like we had reached the "golden age" in using technology in teaching and learning, with multiple mature commercial and open source products available.

The problem with the current VLE marketplace is that significant effort has been invested in products jostling for market share. Each of the vendors watches the others closely, and when one adds a feature that gains a competitive advantage, the others rush to add the same feature to avoid losing market share. Consequently, the products are slowly becoming clones of one another.

FINDING ALTERNATIVES

The most innovative (and fearless) teachers are simply abandoning learning-management systems and using Web 2.0 sites like WikiSpaces, Blogger, YouTube, and Delicious to teach their courses. Teaching outside the VLE is still the domain of a few hardy pioneers who make literacy in Web 2.0 technologies one of the desired learning outcomes for their courses.

Even though Web 2.0 tools are making interesting inroads into the marketplace and revealing that there's a pent-up desire for innovation in the teaching and learning technology marketplace, we haven't yet seen the transformative idea that "changes everything"—the App Store for education.

In many ways, the current VLE market is quite similar to the cell phone market in 2004. Phones then were simple, rugged, and reliable, and highly developed products were available from vendors including Motorola, Nokia, and RIM (BlackBerry). Each phone had whatever features its manufacturer decided to make available—remember Snake from Nokia? Every few years, your phone would wear out or break, and you would get a new one that looked a lot like the old one, with a few more capabilities. In a sense, 2004 was a golden era for the previous generation of cell phones. Both the users and the vendors were quite comfortable with the available offerings.

While the Apple iPhone was certainly innovative in terms of its base technology, its truly transformative aspect was the App Store. Any reasonably competent developer can get an application into the App Store. Apple reviews applications to make sure they're of reasonable quality before putting them into the store, but in general, it lets the market decide which applications are most valuable. More than 350,000 applications are currently available in the App Store.

Interestingly, Google quickly established its own significant market share by building an Android app store following the principles established by Apple.

IMAGINING THE EDUCATION APP STORE

A successful app store needs to be an open channel between the producers of software and content and their consumers. There needs to be a venue where the producers and consumers can find each other, much like a local farmers' market. If these two entities know when and where to show up, and the barriers to entry aren't too high, the market forms naturally.

We've merely scratched the surface of the potential for using technology in teaching and learning.

A technology app store has two critical elements: compliance standards so that users can easily plug in the apps and use them, and marketing and communications between vendors and consumers. In the case of the App Store, the iPhone software development kit and usability guidelines form the standards to ensure that the apps work properly in the iPhone environment.

A consortium of vendors and end users has been developing the IMS Learning Tools Interoperability (LTI) standard (www.imsglobal.org/developers/BLTI), which facilitates plugging apps into learning-management systems. Over the past year, most of the mainstream vendors in the market—Blackboard, Desire2Learn, Sakai, OLAT, Moodle, ATutor, Jenzabar, Instructure, Learning Objects, and others—have either shipped or are in the process of shipping products that support IMS Basic Learning Tools Interoperability. Basic LTI uses a simple OAuth signed message to transfer user identity, course information, and the user role within a course between a VLE and an externally hosted teaching and learning tool. Developers can use Basic LTI to write tools in any language and host them in any kind of environment.

We're waiting for that second critical element—marketing and communications. Some organization needs to take a bit of a risk by building an education app store and lending its name and marketing power to that endeavor.

It's important to keep in mind that Apple didn't have the first or only app store. The first few education app stores might meet with limited success. That isn't a cause for concern, but simply the market sorting itself out. It's also important to realize that there will naturally be more than one successful educational app store in the steady state.

I can see several potential paths to successful educational app stores. A current VLE vendor like Blackboard could create its own app store that's deeply integrated with its products. Publishers like Pearson Education, McGraw-Hill, and Cengage could also build educational app stores, given that it's quite natural to sell both content and software through the same store. Or perhaps the first successful educational app store will be a clever start-up that finds the right niche and overtakes the long-time market participants, bringing untold wealth to its founders. My guess is that ultimately it will be some combination of these scenarios.

It has been almost 15 years since Ron Vetter and I wrote our *Computer* column about two computer scientists experimenting with using the Web as a teaching tool. During that time, the use of technology in teaching and learning has experienced unprecedented growth and success. VLE systems moved from being developed by individual faculty members and running on servers underneath their desks to being the single most important enterprise application on most campuses. We've moved from everyone writing his or her own code to a multibillion-dollar worldwide industry offering a wide range of mature commercial and open source products.

But even with all this progress, we've merely scratched the surface of the potential for using technology in teaching and learning. We're poised to see an explosion of innovations in areas we can't begin to imagine. When we succeed in creating a friction-free marketplace for the exchange of software and content for teaching and learning, we can bring many more creative minds to bear on solving the problem of how to best use this technology in educational settings.

***Charles Severance** is a clinical associate professor in the School of Information at the University of Michigan. Contact him at csev@umich.edu.*

Editor: Ann E.K. Sobel, Department of Computer Science and Software Engineering, Miami University; sobelae@muohio.edu

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

Continued from page 112

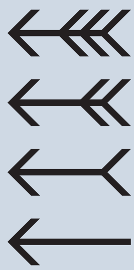


Figure 1. Progressive feathering would offer an option for showing different strengths for arrow keys.

ers, both as consoles used to interact with the operating system, and as terminals used to interact with user programs (tinyurl.com/Wp2741). A little later, a display screen replaced the printer, but the simple typewriter keyboard was still used (tinyurl.com/Chu2260). Interaction was by keying in a command or response and using the carriage return key, nowadays usually labeled Enter, to signal the end of the command or response.

As computers became more complex, in particular in allowing several programs to run at the same time, so did their keyboards. The old laptop PC I'm writing this essay on has 37 text keys and six text control keys (such as Backspace and Shift) that are descended directly from the typewriter keyboard. But there are also 33 other control keys that have their effect on or through the operating system. Most of these other control keys have alternative markings that are a mystery to me.

The idea of a control or function key made sense with a PC, as the user doubles as a system operator. This is probably why a distinct pad with 10 general-purpose control keys to the left of the alphabetic keyboard was introduced with the IBM 5150, as the IBM PC was known at first (tinyurl.com/WpPcKb). These keys were labeled F1 to F10 and had no fixed purpose. Certain conventions were adopted, however, and their general acceptance was followed by their expansion to F12 and their repositioning to a row above the numeric keys on the main part of the keyboard.

Since the function keys were almost always used independently of the numeric keypad, it's a little hard to understand why they weren't overlaid on the numeric keypad with a FnLk key to activate them. There, they would have been easier to use, and there could have been up to 16 on a 4 × 4 numeric keypad.

However, there's a drawback to the function keys: they're arbitrary—their numbering doesn't suggest their significance. In any case, there's a bewildering variety of other control keys and indeed of keyboards generally (tinyurl.com/CmpKbd). What's needed is a standard for keyboards with emphasis on symbolic control keys so that software will work better across cultures.

There's a hint of the possible in the arrow keys that are nowadays found in a ⊥ shaped arrangement usually tucked in near the bottom right of the keyboard. These are cross-cultural ideographs that let the user select a direction of movement: up ↑, down ↓, right →, left ←.

That there are four of these arrow keys suggests that they could be laid over rows of the numeric keypad. With chording, this would allow 15 different functions per row. For example, ↑↓ could select expansion, and →← could select compression. The keypad's four rows could signal different scales or strengths, the higher the stronger, thus giving a total of 60 different functions without using the Ctrl, Alt, or Shift modifier keys. For the down arrow when used to move the cursor in a text file, the four strengths might move the cursor to the next line, the next paragraph, the bottom of the window, and the end of the file, respectively.

The arrows of different strengths would most simply and significantly be shown with progressive feathering, as Figure 1 shows. The Unicode arrow symbols are bewildering in their variety, but feathering seems lacking, and the feathered arrows would need to be added to Unicode for use in documentation (tinyurl.com/WpArSm).

The typical PC user switches between interacting with an individual program, such as a browser in a window, and interacting with the operating system. Therefore, if a SysLk and a PrLk key were provided, presumably in a row of Lk keys above the 4 × 4 keypad proper, 60 functions would be independently available for interacting with the operating system and with the current program before modifier keys would need to be used. This assumes that inter-row chording isn't supported on the grounds of its awkwardness and complexity.

The most important aspect of such an interactive keypad is that it's cross-cultural. The ideographic symbols could be used on a keypad alongside the text keyboard for any writing system, alphabetic or other, and their use could be consistent and memorable across languages.

The allocation of actions to signals would need to be carefully designed and standards agreed upon, but each arrow is capable of various implications. For instance, up and out, down and in, right and forward, and left and back could be accepted couplings. Agreement on a standard would facilitate use of software across cultures, and would simplify and partially unify the design of keyboards. It might also bring some consistency to the multiplicity of interactive devices currently available, and some order to the use of keyboards in cooperation with devices like the mouse and touch screen.

KEYBOARDS AND TEXT

Popular digital technology emphasizes rapid interaction with programs like videogames and browsers. Nevertheless, many people still use it for writing documents of various kinds. This is an important use culturally.

Unfortunately, the digital technology that enables people to interact recreationally with programs has been discouraging them from culturally interacting with text. Of course, text messages over mobile phones and the Internet are flourishing, but this is more superficial and social than cultural.

The problem is the restriction by technology of natural expressiveness. For example, the traditional input methods for Chinese characters involve keying in the pronunciation of a character using an alphabet, then selecting the required character from a list of homophones. However, the typical Chinese character has two parts: a radical that suggests the meaning and a phonetic that suggests the pronunciation. A creative writer might wish to combine any radical with any phonetic, but this isn't possible when the user must select from a list of accepted characters.

For writers of English wishing to be innovative, there's no problem if it's simply a matter of putting together a word or name from the letters available on the keyboard, though a lurking spell-checker might try to interfere. But there's a problem with traditional representations such as résumé, façade, zoölogy, Nº, or CO_2. I can do it in Word, but it isn't easy or obvious for naïve users like me. And I can't think how I might put a right diple (>) over the t in accenture.

There are several reasons for wanting to be able to easily modify English text typographically.

With English becoming more widely used, the use of diacritics and other modifications that make its pronunciation more evident has obvious merit. Being able to better show the pronunciation of foreign names, such as Phœbe, Zhōu Ēnlái, and La Coruña, has merit as well. There's also great potential for the teaching of English (The Profession, June 2006, pp. 102-104).

But the most compelling reason is simply to release imagination and creativity. This reason is becoming culturally more important with the threatened disappearance of handwriting (tinyurl.com/AgJl9w). If children are to be taught to type rather than to write, they need to be able to type creatively, and not just to reduce boredom. The more expressively text can be used, the more appealing literacy becomes.

The obvious way to do this is to lay typographical functions over the numeric keypad with a TyLk key above it to enable typographic control while keying in text. The arrow symbols could be well suited for this, but they would need to be used in conjunction with the plaintext keyboard. Much could be done—in particular, belated restoration of the very useful backspace/overtype capability of traditional typewriters that the computing profession did away with.

The possibilities go way beyond simply the 60 keypad functions available. For example, by holding down a key of the alphabetic keyboard, the keypad could be used to modify the letter being keyed in, making it italic and wider, and rotating and reversing it. By holding down a typographic key, a sequence of letters could be keyed in to combine them, for example, marking diacritically and overlaying or ligating (tinyurl.com/WpTyLg).

These examples are relevant to cultures that use the Latin alphabet. Cultures that use other writing systems would need different enhancements for rich text, but the SysLk and PrLk controlled interaction would be able to cross writing systems.

The point of all this is that digital technology could greatly enhance both the interactive and textual use of computers by developing and standardizing keyboard technology. The computing profession has a social responsibility to press for this to be done and to see that the software exploits the extra capability thus provided, especially for schools. ■

***Neville Holmes** is an honorary research associate at the University of Tasmania's School of Computing and Information Systems. Contact him at neville.holmes@utas.edu.au.*

Computer | Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page | Qmags THE WORLD'S NEWSSTAND®

The Cultural Potential of Keyboards

Neville Holmes,
University of Tasmania

Culture has many facets and, with modern digital technology, many opportunities.

Human culture has been dependent on digital systems for a very long time, a digital system being one that "uses discrete (discontinuous) values" (tinyurl.com/WkPdDg).

Human language, the main enabler of our cultural development, has developed digitally. Spoken language is a combination of analog intonation and digital phonemes. Written language is notionally digital, and printed language is thoroughly so. Alphabetic language uses letters to indicate phonemes, punctuation to occasionally suggest intonation, and logograms such as & and 7 to represent words.

Musical culture also has a digital basis. For example, the keyboard was invented for a water organ more than 2,000 years ago (tinyurl.com/WpHdrls). Such keyboards are linear, suited to selection from a wide row of pipes or strings, but the keyboards developed much later for typewriting and typesetting were bidimensional because they were used for building narrow columns of words.

The typesetting keyboard arranged the letters of the alphabet in sequence by their frequency of use, so the first two rows, etaoin and shrdlu, became famous in the days of hot metal printing (tinyurl.com/WpEtSh). The typewriter keyboard was laid out rather peculiarly to avoid mechanical malfunction, hence the qwerty sequence (at least for the English language) and the staggered placement (tinyurl.com/WpKbLo).

In the days of 6-bit bytes with letters only uppercase, simple teletype keyboards were used with computers. When 8-bit bytes—EBCDIC and ASCII—were introduced during the 1960s, fairly straightforward typewriters were used as consoles and terminals.

Below, I include links to equipment I used myself, although quite a variety was available.

KEYPADS AND NUMBERS

Early PCs were expensive and often seen as glorified calculators. This was probably why many of them had a numeric keypad added to the alphabetic keyboard. An example was the IBM 5100 released in 1975 that had a very complex keyboard, including a numeric keypad, and a variety of control keys. It had a rather small screen but nonetheless was useful when lecturing because it had a video outlet for large-screen display.

The numeric keypad on the alphabetic keyboard was very popular because early commercial PCs were mainly used in offices, either for word processing or for spreadsheets and other calculations.

The keypad, usually of a 4 × 4 or similar design, with the rightmost column used to incite simple arithmetic, was much more suited to calculation than placing the row of numeric keys above the alphabetic keys, with its 0 strangely following the 9. Keypad use is still seen in shops with PCs coupled to cash drawers.

More widespread use of PCs and their descendants has reduced the need for simple calculation and increased the need for smaller size. Thus, the keypad is often superimposed over the right-hand side of the alphabetic keyboard with a (rarely used) NumLock key to bring it into play. Curiously, my cordless phone has gone the opposite way for compactness and lays the alphabet over a 4 × 3 numeric keypad.

Of course, the numeric subculture is unpopular today, but its keypad is a model of simplicity. Numbers can be keyed in with minimal hand movement, and touch-keying is relatively easy to learn.

KEYBOARDS AND PROGRAMS

Before PCs, fairly straightforward typewriters were coupled to comput-

Continued on page 110

Published by the IEEE Computer Society

Computer Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page Qmags THE WORLD'S NEWSSTAND®


Focus on
Your Job Search
IEEE Computer Society Jobs helps you easily find a new job in IT, software development, computer engineering, robotics, programming, artificial intelligence, networking and communications, consulting, and other computer-related areas.
Visit www.computer.org/jobs to search technical positions from employers worldwide.
http://www.computer.org/jobs
The IEEE Computer Society is a partner in the AIP Career Network, a collection of online job sites for scientists, engineers, and computing professionals. Other partners include Physics Today, the American Association of Physicists in Medicine (AAPM), American Association of Physics Teachers (AAPT), American Physical Society (APS), AVS Science and Technology, and the Society of Physics Students (SPS) and Sigma Pi Sigma.