



Innovation and Obstacles: The Future of Computing

In this multidisciplinary glimpse forward, some of this decade's key players offer opinions on a range of topics—from what has driven progress, to where innovation will come from, and to obstacles we have yet to overcome.

Moderator:

David D. Clark
Massachusetts Institute of Technology, Chair of the Computer Science and Telecommunications Board

Panelists:

Edward A. Feigenbaum
US Air Force

Juris Hartmanis
Cornell University

Robert W. Lucky
Bellcore

Robert M. Metcalfe
International Data Group

Raj Reddy
Carnegie Mellon University

Mary Shaw
Carnegie Mellon University

In this excerpt from “Visions for the Future of the Fields,” a panel discussion held on the 10th anniversary of the US Computer Science and Telecommunications Board, experts identify critical issues for various aspects of computing. In the accompanying sidebars, some of the same experts elaborate on points in the panel discussion in mini essays: David Clark, CSTB chairman, looks at the changes needed in computing science research, Mary Shaw of Carnegie Mellon University examines challenges for software system designers, and Robert Lucky of Bellcore looks at IP dialtone, a new infrastructure for the Internet. Donald Greenberg of Cornell University rounds out the essays with an outlook on computer graphics. Finally, in an interview with William Wulf, president of the US National Academy of Engineering, *Computer* explores the roots of innovation and the broader societal aspects that will ultimately drive innovation in the near term.

RECKLESS PACE OF INNOVATION

Clark: We have heard the phrase “the reckless pace of innovation in the field.” I have a feeling our field has just left behind the debris of half-understood ideas in an attempt to plow into the future. Do you think we are going to grow up? Ten years from now, will we still say we have been driven by the reckless pace of innovation? Or will we, in fact, have been able to breathe long enough to codify what we have actually understood so far?

Reddy: We have absolutely no control over the pace of innovation. It will happen whether we like it nor not. It is just a question of how fast we can run with it.

Lucky: At Bell Labs, we used to talk about research in terms of 10 years. Now you can hardly see two weeks ahead. The question of what long-term research is all about remains unanswered when you cannot see what

is out there to do research on. Nicholas Negroponte was saying recently that, when he started the Media Lab at the Massachusetts Institute of Technology, his competition came from places like Bell Labs, Stanford University, and the University of California at Berkeley. Now he says his competition comes from 16-year-old kids. I see researchers working on good academic problems, and then two weeks later some young kids in a small community are out there doing it. There must still be good academic fields where you can work on long-term problems in the future, but the future is coming at us so fast that I just sort of look in the rear-view mirror.

Shaw: I think innovation will keep moving; at least I hope so, because if it were not moving this fast, we would all be really good IBM 650 programmers by now. What will keep it moving is the demand from outside. We have just begun to get over the hump where people who are not in the computing priesthood, and who have not invested many years in figuring out how to make computers do things, can actually make computers do things. As that becomes easier—it is not easy yet—more and more people will be demanding services tuned to their own needs. They will generate the demand that will keep the field growing.

Hartmanis: We can project reasonably well what silicon technology can yield during the next 20 years; the growth in computing power will follow the established pattern. The fascinating question is, what is the next technology to accelerate this rate and to provide the growth during the next century? Is it quantum computing? Could it really add additional orders of magnitude? What technologies, if any, will complement and/or replace the predictable silicon technology?

Clark: Are growth and demand the same as innovation? We could turn into a transient decade of interdisciplinary something, but does that actually mean there is any innovation in our field?

Shaw: We have had some innovation, but it has not been our own doing. Things like spreadsheets and word processors, for example, have started to open the door to people who are not highly trained com-

Excerpts of “Visions for the Future of the Fields,” *Defining a Decade: Envisioning CSTB's Second 10 Years*, ©1997 are reprinted with permission by the National Academy of Sciences. Courtesy of the National Academy Press, Washington, DC.

puting professionals, who have come at the academic community from the outside. I remember when nobody would listen if you wanted to talk about text editors in an academic setting. Most recently, there has been the upsurge of the World Wide Web. It is true that Mosaic was developed in a university, but not exactly in the computer science department. These are genuine innovations, not just nickel-and-dime things. **Feigenbaum:** I think the future is best seen not in terms of changing hardware or increased numbers of MIPS (or GIPS), but rather in terms of the software revolution. We are now living in a software-first world. I think the revolution will be in software building that is now

done painstakingly in a craftlike way by the major companies producing packaged software. They create a suite—a cooperating set of applications—that takes the coordinated effort of a large team. What we need to do now in computer science and engineering is to invent a way for everyone to do this at his or her desktop; we need to enable people to “glue” packaged software together so that the packages work as an integrated system. This will be a very significant revolution.

I think the other revolution will be the one Leonard Kleinrock called *didactic* or *intelligent agents*. The agent allows you to express what you want to accomplish, providing the agent with enough knowledge

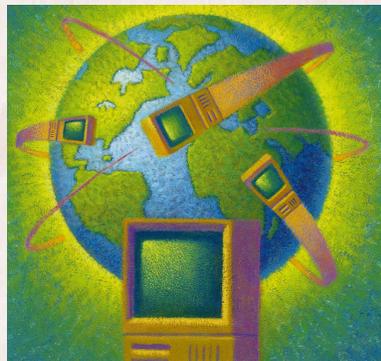
Outlook on Computer Science

Ten years from now, we may conclude that this was the coming of age for computer science. Certainly, it is going through a transition, which like many transitions, may appear a little painful to some living through it.

It's a lot like the fencing of the American West. Those who worked in the early decades of CS had wide-open spaces for original ideas to flourish. That space is now populated with mature innovations, successful products, and large corporations, which I'm sure seems confining to some. The current context of CS is shaped by past success and chronic trouble spots.

Past success makes blue-sky innovation seem more daunting and risky. It can trap unwary researchers in the present and keep them from exploring the future. What innovative operating system could have any impact, given the market presence of Microsoft? What novel idea for global networking could displace the Internet? What new paradigm of computing could compete with the PC?

At the same time, some of the intellectual problems CS has struggled with, such as building large and trustworthy systems, seem to have a timeless quality: The complaints about building systems seem the same as a decade ago. But there *has* been progress—in the last decade networks have interconnected the world and made a new



range of large systems possible. Perhaps the real issue is that aspirations to build bigger and more complex systems are potentially unbounded, kept in check only by the limits of what can actually be built. So system builders live in a constant state of frustration, always hitting (and complaining about) the same limitations, even though the systems they can now build are indeed bigger and more complex than a decade ago.

Past successes change the imperative for future research. Whereas the past was defined by articulating new objectives (Let's build an Internet), past achievements now demand incremental innovation (Let's add support for audio streams to the Internet). That sort of work, while less broad in scope, is critical to the progress of the field. Part of coming of age is learning to equally respect innovation *and* enhancement.

The locus of the hard CS problems has also shifted. Look at the systems area. Although there's still room for innovation in traditional areas such as language design (consider Java), more and more of the hard problems arise when people try to put computers to use. This implies that if com-

puter scientists are to contribute to important emerging problems, they must increasingly act as anthropologists and go live for a time in the land of the application builders. Computer scientists are now working on diverse problems: ensuring privacy, rendering complex images for realistic games, and understanding human speech. The Web is a wonderful example of an application that has spawned lots of interesting problems, such as the construction of powerful search engines, algorithms for caching and replicating Web pages, the naming of information objects and secure networks.

Of course, parts of CS view their role less as innovating new artifacts and more as developing the underpinnings of CS as a science. There is much about CS that is not yet understood in any rigorous way, and the rate of innovation turns up new problems to understand faster than the old ones can be solved. For those who build the foundations of the field, there is no immediate fear of running out of problems to work on, even if innovation were to slow down.

Does the coming of age mean that the era of big change is over? I think not. In the next 10 years, user interfaces, the PC, the Internet, and even Windows will have mutated, perhaps almost beyond recognition. What will actually characterize the next decade is the sometimes turbulent interplay between improving the past and overthrowing it at the same time. Anyone who thinks the fun is over has just walked off the field at half time. ♦

— David D. Clark, Chair, Computer Science and Telecommunications Board

about your environment and your context for it to reason exactly how to accomplish it.

SILICON AND FRUSTRATION

Clark: One statement made at the beginning of this decade was that the nineties would be the decade of standards. There is an old joke: the nice thing about standards is that there are so many to pick from. In truth, I think that one of the things that has happened in the nineties is that a few standards happened to win some sort of battle—and not because they are necessarily the best.

Lucky: This is both a tragedy and a great triumph. You can build a better processor than Intel or a better operating system than Microsoft, but it does not matter. It just does not matter.

Clark: How can you hurtle into the future at a reckless pace and simultaneously conclude that it is all over, it does not matter because you cannot do something better, because it is all frozen in standards?

Metcalf: There seems to be reckless innovation on almost all fronts except two, software engineering and the telco monopolies.

Clark: Yet if we look at the Web, we have a regrettable set of de facto standards in HTML and HTTP, both of which any technologist would love to hate. When you try to innovate by saying it would be better if URLs were different, the answer is, “Yes, well there are only 50 million of them outstanding, so go away.” Therefore, I am not sure I believe your statement that there is rapid innovation everywhere, except for these two areas.

Lucky: It is possible that if all the dreams of the Java advocates come true, this will permit innovation on top of a standard. It is one way to get at this problem. We do not know how it is going to work out, but at least this would be the theory.

Clark: I actually believe it might be true. A tremendous engine exists that is really driving the field and that is the rate of at least performance innovation, if not cost reduction, in the silicon industry. I think this engine drove us forward, but I am not sure it's the only engine. I wonder if [ten years from now] we will say, “Yes, silicon drove us forward,” or will there be other things? Is the Web a creation of silicon innovation?

Shaw: No, it is a creation of frustrated people who did not feel like dealing with ftp and telnet, but still wanted to get to information.

Clark: I think you just said that silicon and frustration are our drivers.

Lucky: Silicon has really made everything possible. This is undeniable, even though we spend most of our time, all of us, working on a different level.

Clark: I once described setting standards on the Internet as being chased by the four elephants of the apocalypse. The faster I ran, the faster they chased me because the only thing between them and making a

billion dollars was the fact that we had not worked this thing out. We cannot outrun them. If it is a hardware area, we can hallucinate something so improbable we just cannot build it today. Then, of course, we cannot build it in the lab, either. We used to try to have hardware that let us live 10 years in the future. Now I am hard-pressed to get a PC on my desk. Yet in the software area, there really is no such thing as a long-term answer. If you can conceive it, somebody can reduce it to practice. So I do not know what it means to be long term anymore.

Feigenbaum: If you look at what individual faculty people do, you find smallish things in a world that seems to demand more team and system activity. There is not much money around to fund anything more than small things, basically to supplement a university professor's salary and a graduate student or two, and perhaps run them through the summer. Partly this is because of a general lack of money. Partly it is because we have a population explosion problem and all these mouths to feed. All the agencies that were feeding relatively few mouths 20 years ago are now feeding maybe 100 times as many assistant professors and young researchers, so the amounts of money going to each are very small. This means that, except for the occasional brilliant meteor that comes through once in a while, you have relatively small things being done. When they get turned into anything, it is because the individual faculty member or student convinces an industry to spend more money on it. Subsequently, the world thinks it came out of industry.

LOOKING OUTWARD AT REAL PROBLEMS

Anita Borg (audience member): I wanted to talk a bit about where you get innovation and where academics get ideas for problems to work on. This is something I talk about every time I go, as an industry person, to talk to a university. If we keep training students to look inside their heads and become professors, we lose the path of innovation. If we train our students to look at what industry is doing and what customers and people out there using these things cannot do—to not be terrorized by what they can do, but to look at where they are running into walls—our students start appreciating these as the sources of really hard problems. I think this focus is lacking in academia to some extent and looking outward at real problems gives you focus for research.

Hartmanis: I fully agree. Students should be well aware of what industry is and is not doing, and I believe that many are well informed.

Shaw: Earlier I mentioned three innovations that came from outside the computer science community: spreadsheets, text formatting, and the Web. I think they came about because people outside the community had

Someone once said the nineties would be the decade of standards. There is an old joke: The nice thing about standards is that there are so many to choose from.

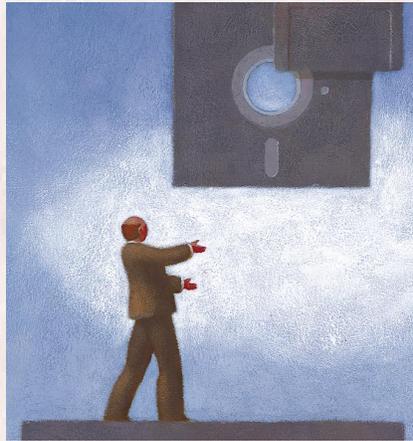
Outlook on Software System Design

Over the last decade, computers have become nearly ubiquitous, and their users are often people who neither have, nor want, nor (should) need, years of special training in computing. Business computers are often in the hands of information users, no longer under exclusive control of a centralized information systems department. Instead of gaining access to computers only through professional intermediaries, vast numbers of people are responsible for their own computing—often with little systematic training or support. This *disintermediation*—the direct association between users and their software—has created new problems for software system designers.

If all these computers are to be genuinely useful, their owners or handlers must be able to control them effectively; they must understand how to express their problems, they must be able to set up and adapt the computations, and they must have reasonable assurance of the correctness of their results. This must be true across a wide range of problems, spanning business solutions, scientific and engineering calculations, and document and image preparation. Furthermore, owners of personal computers must be able to carry out the tasks formerly delegated to system administrators, such as configuration, upgrade, backup, recovery.

The means of disintermediation have been available, affordable hardware together with application-targeted software that produces information and computing in a form the end user can understand and control. The software carriers—the “killer apps”—have been spreadsheets, the Web, integrated office suites, and interactive environments such as MUDs (multiuser domains). To a lesser degree, Visual Basic has enabled people with minimal programming experience to create useful software that fits their own needs. These applications have become winners in the marketplace because they put a genuinely useful capability in the hands of people with real problems.

The software system design community



should be alarmed to notice that these killer apps have emerged from outside their research world. Worse, the research community has often (at least initially) failed to take these applications seriously. Such applications have been regarded as toys, not worthy of serious attention; they have been faulted for a lack of generality or (imagined) lack of scalability; they have been ignored because they don't fit the established research mold. But software system design researchers comprise the very community that should be breaking the mold and providing solutions for real-world needs.

Although it's always risky to predict the market, one place to look for ideas is the relation between the people who use computing and the computing they use. We've already seen substantial disintermediation, as more and more people have direct access to their computers and software. At present, their computing is dominated by individual interactive computations, which lets them monitor results as they go. It is more challenging to set up stand-alone processes that run unmonitored. This requires describing policy for an open-ended set of computations, not just manipulating instances. We can see the small beginnings of such independent processes in mail filters, automatic check payments, and the daemons that select articles from news feeds. But what will be required to enable large numbers of users to set up autonomous software agents with larger responsibilities? At what point will the public at large trust the Internet and electronic commerce mechanisms enough to carry out individual transactions? When will consumers be willing to have an autonomous software agent spend money on their behalf?

Another potential change in the relation between people and computing is a fusion between computing and entertainment. This will, of course, require infrastructure development in bandwidth, 3D display, intellectual property protection, and electronic commerce—the usual stuff of software system design research. Beyond that, though, what new capabilities will the consumer need? What will be required to make entertainment both interactive and multiparty? How can individuals become producers as well as consumers of computer-based entertainment?

What does all this mean for software system design research? First, we must recognize the important—and difficult—research problems that these applications carry, including how to

- analyze component interoperability and develop techniques for coping with incompatibility;
- specify and implement event-driven systems that support the dynamic reconfiguration of loosely confederated processes or agents;
- support metainformation that carries type, signature, performance, and other information needed to automate distributed agents;
- manage families of related systems;
- deal with the security issues of electronic commerce;
- design for “gentle-slope systems,” in which the learning time required is commensurate with the application's sophistication;
- integrate multiparty real-time interaction with other applications (beyond chat rooms, electronic whiteboards, MUDs, and virtual communities); and
- analyze requirements for market segments rather than individual bespoke systems.

Second, we should contribute to developing accurate models of computer use that are simple enough for nonspecialists to understand. Finally, we should increase interdisciplinary work with researchers in human-computer interaction and in application areas. ♦

—Mary Shaw, Carnegie Mellon University

something they needed to do and were not getting any help doing it. So we will get more leads by looking not only at the problems of computer scientists, but also at the problems of people who do not have the technical expertise to cope with these problems. I do not think the next innovation is going to be an increment along the Web, or an increment on spreadsheets, or an increment on something else. What Anita is asking us to think about is, how are we going to be the originators of the next killer application, rather than waiting for somebody outside to show it to us?

Reddy: If you go back 40 years, it was clear that certain things were going to have an impact on society—for example, communications satellites, predicted by Arthur Clarke; the invention of the computer; and the discovery of the DNA structure. At the same time, none of us had any idea of semiconductor memories or integrated circuits. We did not conceive of the Arpanet. All of these came to have an impact. So my hypothesis is that some things we now know will have an impact. One is digital libraries. The term digital library is a misnomer, the wrong metaphor. It ought to be called digital archive, bookstore, and library. It provides access to information at some price, including no price. In fact, the National Science Foundation and DARPA have large projects on digital libraries, but they are mainly technology-based—creating that technology to access information. Nobody is working on the other problem of content.

We have a Library of Congress with 30 million volumes; globally, the estimate is about 100 million volumes. The US Government Printing Office produces 40,000 documents consisting of six million pages that are out of copyright. Creating a global movement—because it is not going to be done by any one country or any one group—to get all the content (to use Jefferson's phrase, all the authored works of mankind) online is it. At Carnegie Mellon University, we are doing two things to help. In collaboration with National Academy Press, we are beginning to scan, convert, correct, and put in HTML format all its out-of-print books. There are already about 200 to 300 of them. By the end of the year, we expect to have all of them. The second thing CMU is doing is offering to put all authored works of CSTB members on the network.

FIXING THE INTERNET

Metcalfe: The Internet is one of our big success stories and we should be proud of it, but it is broken and on the verge of collapse. It is suffering numerous brownouts and outages. Increasingly, the people I talk to, numbering in the high 90 percent range now, are generally dissatisfied with its performance and reliability.

There is no greater proof of this than the proliferation of intranets. The good reason people build them is to serve internal corporate data processing applica-

tions, as they always have. The bad reason they build them is that the Internet offers inadequate security, performance, and reliability for its uses. So we now have a phenomenon in companies. The universities, as I understand it, are currently approaching NSF to build another NSFnet for them. This is really a suggestion not to fix the Internet, but to build another network for us.

Of course, the Internet service providers are also tempted to build their own copies of the Internet for special customers and so on. I believe this is the wrong approach. We need to be working on fixing the Internet. Lest you be in doubt about what this would include, it would mean adding facilities to the Internet by which it can be managed. I claim that these facilities are not in the Internet because universities find management boring and do not work on it.

Fixing the Internet also would include adding mechanisms for finance so that the infrastructure can be grown through normal communication between supply and demand in our open markets, and adding security; it is not the National Security Agency's fault that we do not have security in the Internet. It occurred because for years and years working on security has been boring, and no one has been doing it; now we finally have started.

We also need to add money to the Internet—not the finance part I just talked about, but electronic money that will support electronic commerce on the Internet. We need to introduce the concept of zoning in the Internet. The Communications Decency Act is an effort, although a lame one, to bring this about. On the Internet, mechanisms supporting freedom of speech have to be matched by mechanisms supporting freedom not to listen.

We need progress on the development of residential networking. The telecommunications monopolies have been in the way for 30 or 40 years, and we need to break these monopolies and get competition working on our behalf.

Shaw: We talked a lot about software and a little about the Web, which is really a provider of information rather than of computation at this point. I believe we should not think about these two things separately, but rather about their fusion as information services, including not only computation and information, but also the hybrid of active information. On the Web, we have lots of information available as a vast undifferentiated sea of bits. We have some search engines that find us individual points. We need mechanisms that will allow us to serve more systematically and to retain the context of the search. To fundamentally change the relation between the users and the computing, we need to find ways to make computing genuinely widespread and affordable, pri-

We need to fix the Internet, which means adding facilities by which it can be managed. These facilities are not in the Internet because universities find management boring and do not work on it.

vate and symmetric, and genuinely intellectually accessible by a wider collection of people.

I thank Bob Metcalfe for saying most of what I was going to say about what needs to be done because the networks must become places to do real business, rather than places to exchange information among friends. In addition, we need to spend more time thinking about what you might call naïve models, that is, ways for people who are specialists in something

other than computing to understand the computing medium and what it will do for them, and to do this in their own terms so they can take personal control over their computing.

Lucky: I know two things about the future. First, after the turn of the century, one billion people will be using the Internet. Second, I do not have the foggiest idea what they are going to be using it for. We have created something much bigger than us, where biological rules

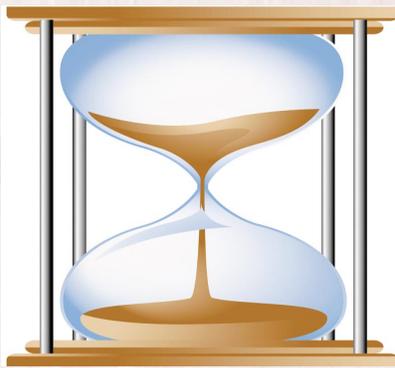
Outlook on Telecommunications

The familiar telephone network has served us well for a century. Yesterday's grand challenge—connecting the planet with voice telephony—has been accomplished.

But as the new century looms, a revolutionary model for telecommunications has thrust itself on an unsuspecting industry. It is a model based on the Internet, where all communications take the form of digital packets, routed from node to node according to IP, or Internet Protocol. This is a world in which the Esperanto that enables intercommunication between disparate networks means expressing everything in IP packets. The future plug on your wall will speak IP, and the service it offers will be IP dial tone.

Today's infrastructure is a circuit-switched network in which the dominant traffic is voice, and the medium of exchange among networks is the standard 3-kHz analog channel. To transmit data, we use modems to change the digital signal into a voicelike analog signal compatible with this transmission format. The network of the future will invert this paradigm—reformatting voice to look like data. The natural medium of exchange will be the IP packet.

The new IP-dial tone network is happening very fast. It is well known that the number of host computers on the Internet doubles annually, and estimates on the annual growth of data traffic in the Internet backbone start at a factor of 4 and go up to 10. Thus the traffic is growing faster than the number of users, indicating



that the average user is consuming considerably more bandwidth each year.

People speak of two forthcoming events: The *crossover*, when data traffic equals voice traffic, will probably happen in the next several years. Soon after, however, we will have the *eclipse*, when data traffic becomes an order of magnitude larger than voice traffic.

Moreover, if data traffic is indeed growing by an annual factor of 10, the eclipse could be sudden. Almost immediately after data traffic pulls equal to voice, it will begin to supplant voice as the dominant traffic on the world's networks. And we would essentially have an IP dial tone network.

To many engineers, sending voice as data packets has sounded awkward, even silly. Voice is, after all, continuous and analog by its nature. Why break it into packets with varying and unreliable arrival times? There are several good reasons. The first and most immediate is that Internet telephony (voice on IP) is essentially free. Even if this somewhat artificial advantage does not survive, more sustainable potential advantages include the integration of multimedia content, embedded signaling, and the integration of the computer and telecommunications environments on the desktop. Transmission efficiency may also improve somewhat

because of better speech coding and multiplexing.

The new IP-dial tone network portends not just a revolution in technology, but in the very basis of telecommunications economics. Today the cost of the network and its operation are supported by charging fees based on call time and distance. The tariffs have historically been set according to the guiding principle of universal service. In the IP world of packets it is not at all clear what might constitute a natural basis for charging. Moreover, the technology revolution is being accompanied by a complete restructuring of the regulatory environment. The system is being unbundled, and the infrastructure will be provided by many competing service providers, each with a rich choice of technology alternatives. The only constant holding together these disparate elements will be the IP.

The IP world has often been viewed as an hourglass. On the wide top are the applications; on the bottom are all the alternative physical transmission technologies. The narrow waist is the IP. Anything that speaks IP can flow through the waist unimpeded. That narrow waist isolates the myriad complexities of the underlying world from the equally daunting complexities of the upper applications. That is the real beauty in the retrospective appreciation of the IP.

In an IP world, the user is empowered to build applications on a minimally defined standard. We have already seen overwhelming evidence of that empowerment in the Internet. New applications continually spring from all corners of the world. There is a magic in the air of Internet, and it will soon emanate from that little IP plug in the wall. Welcome to IP dial tone. ♦

—Robert W. Lucky, Bellcore

seem more relevant than the future paradigm we are used to, where Darwinism and self-adaptive organization may be the more relevant phenomena with which to deal. The question is, How do we design an infrastructure in the face of this total unknown? Certain things seem to be an unalloyed good that we can strive for. One is bandwidth. Getting bandwidth out all the way to the user is something we can do without loss of generality.

On the other side, it is hard to find other unalloyed goods. For example, intelligence is not necessarily a good thing. Recently there was a flurry of e-mail on the Internet when one of the router companies announced that it was going to put an “Exon box” in its router. An Exon box would check all packets going by to see if they are adult packets or not. There was a lot of protest on the Internet, not because of First Amendment and Communication Decency Act principles, but because people did not want anything put inside the network that exercises control, simply as an architectural paradigm, more than anything else. So bandwidth is good, but anything else you do on the network may later come back to bite you because of profound uncertainty about what is happening.

LIMITS OF RATIONAL REASONING

Hartmanis: I would like to talk more about the science part of computer science, namely, theoretical work in computer science and its relevance, and identify some stubborn intellectual problems. For example, security and trust on the Internet are of utmost importance; yet all the methods we use for encryption are based on unproven principles. We have no idea how hard it is to factor large integers, but our security systems are largely based on the assumed difficulty of factoring. There are many more such unresolved problems about the complexity of computations that are directly relevant to trust, security, and authentication, as well as to the grand challenge of understanding what is and is not feasibly computable. The notorious $P=NP$ is probably the best-known problem of this type, but by far not the only one. I consider these among the most important problems in theoretical computer science and sincerely hope that, during the next 10 years, some of them will be solved. I believe that deeper understanding of the computing paradigm, the quest to understand what is and is not feasibly computable is equivalent to understanding the limits of rational reasoning—a noble task indeed.

LIGHTING THE WORLD

Feigenbaum: I would like to talk briefly about artificial intelligence and the near future. If we look back 50 years—in fact to the very beginning of computing—Turing was around to give us a vision of artificial intelligence and what it would be, beautifully explicated in the play about Turing’s life, *Breaking the Code*.

Raj Reddy published a paper in the May 1996 *Communications of the ACM*, his Turing Award address, called “To Dream the Possible Dream.” I share that possible dream, but I feel like the character in the William Steif cartoon who is tumbling through space saying, “I hope to find out what it is all about before it is out.”

There is a kind of Edisonian analog to this. Yes, we have invented the light bulb, and we have given people plans to build the generators. We have given them tools for constructing the generators. They have handcrafted a few generators. There is one lamp post working here, or lights on one city block are working over there. A few places are illuminated, but most of the world is still dark. Yet the dream is to light up the world! Edison, of course, invented an electric company. So the vision is to find out what it is we must do—and I am going to tell you what I think it is—and then go out and build that electric company.

What we learned over the last 25 years is that the driver of the power of intelligent systems is the knowledge the systems have about their universe of discourse, not the sophistication of the reasoning process the systems employ. We have put together tiny amounts of knowledge in very narrow, specialized areas in programs called *expert systems*. These are the individual lamp posts or, at most, the city block. What we need built is a large, distributed knowledge base. The way to build it is the way the data space of the Web came about—a large number of individuals contributing their data to the nodes of the Web. In the case I am talking about, people will be contributing their knowledge in machine-usable form. The knowledge would be presented in a neutral and general way—a way to build knowledge bases so that they are reusable and extensible, so that the knowledge can be used in many applications. A lot of basic work has been done to enable this kind of infrastructure growth. I think we just need the will to go down that road. ❖

Acknowledgments

Computer thanks Nancy Talbert for soliciting and coordinating the Outlook sidebars and David Clark for reviewing the excerpted panel discussion.

David D. Clark is a senior research scientist at Massachusetts Institute of Technology’s Laboratory for Computer Science and chair of the CSTB. Contact him at ddc@lcs.mit.edu.

Edward A. Feigenbaum is chief scientist of the US Air Force. Contact him at feigenbaum@hq.af.mil.

Donald P. Greenberg is Jacob Gould Schurman pro-

Outlook on Computer Graphics

As we continue our inexorable quest for realistic image synthesis in real time, we can at last say that our goal is close at hand. Indications are that by 2025, we will have the technology to produce realistic real-time images at resolutions up to or beyond the limits of our visual perception. The algorithms used will be pixel-based, not polygon-based, and the images will include all the subtle effects of shading, shadows and interreflections we see in complex environments. What this means is that we will be able to create images that are visually indistinguishable from real-world scenes.

How will we get such simulations? I see progress as having three phases.

The first phase deals with local light reflection. We now have the algorithms to model physically based reflection behavior of light scattering off a surface. The models result in what is standardly termed the diffuse, directional diffuse, and specular components. Researchers are currently extending these models to include fluorescence, polarized light, and multilayered materials.

The next phase deals with the propagation of light energy throughout the environment. Because each surface can interact with every other surface in receiving or distributing radiant energy, this problem is complex. The accuracy of the solution depends on how the environment is dis-



cretized for the simulation, the original geometric complexity, and the precision of the computational solution. Computation times for these simulations are so severe that in today's computing environment, only static scenes are practical. These tasks might require 10^{10} times more processing power than we currently have today on a single workstation. Fortunately, much of the calculation has no visual effect on the resulting image, so we need not calculate beyond the limits of human perception. This would require only 10^7 times more processing power.

The third phase involves establishing tone-mapping procedures, to map the physical predictions to within the limits of the display devices. Displays are constrained by their spatial resolution (number of pixels), color resolution (number of color channels), dynamic range (number of illumination levels), and temporal resolution

(number of frames per second). Today's display systems, even high-resolution monitors, are not sufficient. High-definition television comes closer, but another factor of 10 in some dimensions is probably necessary. This display capability will probably be reached within the next two decades.

So if the display devices are available by 2020, will we be able to compute the simulations in real time? According to Moore's law (chip densities double every 18 months), in 15 years we will have 10^6 times more processing power than today, and within five years after that, approximately 10^7 times. More radical predictions, which include the shift from aluminum to copper circuitry, the use of multi-bit transistors, and the inclusion of parallelism, claim that Moore's law will even be exceeded.

This means that sometime near the end of 2025 we will have both the display and computational capability to produce images that are both physically accurate and perceptually indistinguishable from real-world scenes.

If this comes true, we then face another dilemma. The good news is that our communication capability will be vastly enhanced. The bad news is that seeing is no longer believing. Images may no longer be admissible as evidence, and verification tools might be needed to avoid confusion between real and virtual worlds. But I don't expect these cultural issues to slow our quest for realism in image synthesis. Indeed, a picture will be worth 1,024 words. ♦

— Donald P. Greenberg,
Cornell University

Professor of computer graphics at Cornell University. Contact him at dpg@graphics.cornell.edu.

Juris Hartmanis is the National Science Foundation's assistant director for computer and information science and engineering. Contact him at jhartman@nsf.gov.

Robert W. Lucky is corporate vice president of applied research at Bellcore. Contact him at rlucky@bellcore.com.

Robert M. Metcalfe is executive correspondent for

Infoworld and vice president of technology at International Data Group. Contact him at metcalfe@infoworld.com.

Raj Reddy is dean of Carnegie Mellon University's School of Computer Science and the Herbert A. Simon University professor of computer science and robotics. Contact him at raf.reddy@cmu.edu.

Mary Shaw is the Alan J. Perlis professor of computer science and associate dean for professional programs at Carnegie Mellon University. Contact her at mary.shaw@cs.cmu.edu.

William Wulf on the Many Faces of Innovation

Interviewed by Nancy Talbert

Computer: Every significant change seems to be associated with a push and pull. What are the next major pushes and pulls?

Wulf: I wish I could say otherwise, but I think the major push has been and will continue to be Moore's law. Anything that is changing at that rate just can't be ignored. We don't have an equivalent law for bandwidth, but we are obviously getting an even greater rate of change there. I think those fundamental hardware advances will drive everything else. The pulling forces are a lot more diffuse and harder to see. We've got a phenomenon for which we have no real precedent. The first time I saw a spreadsheet, for example, I did not appreciate the impact they would have, because there had never been a thing like them before, at least not mechanized. Because we have less experience, I think it is very much harder to identify the pulls.

Computer: And therefore much harder to say where innovation will come from?

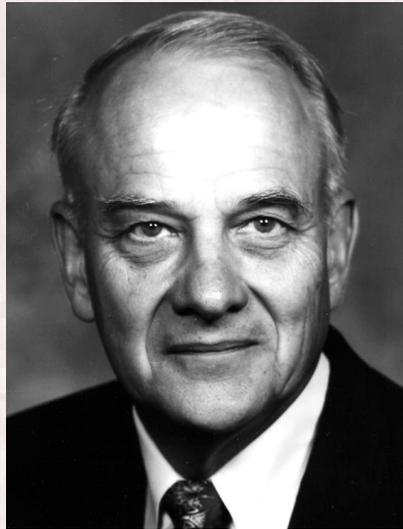
Wulf: Exactly. We are lucky in this country to have the atmosphere to grow more Thomas Edisons—groups of entrepreneurs that will generate the pull. Who knows were the next light bulb will come from?

Computer: How will computers evolve?

Wulf: I suspect that computers will simply become parts of products that we will view as having some intelligence, some responsiveness to human needs. My car, I am told, has seven microprocessors in it, but I am not conscious of them. It will be interesting to see whether there is an identifiable thing called a computer 20 years from now. My guess is there won't be.

Computer: A year ago, you stated "Interesting and deep academic problems are spawned by short-term product development." What do you think will be the next major product-problem set?

Wulf: One example is MEMS [microelectromechanical systems] technology. MEMS lets the designer put on a small piece of silicon something that can sense its environment, reason about the implications of the sensory information, plan what to do about it, physically act, and communicate its actions. Although we don't talk about it much, there's no reason we can't



also put radio transmitters and receivers on MEMS devices. It seems to me that this opens enormous possibilities. Every one of these devices is going to create a complex system or be embedded in one. I have infinite faith that that will generate a set of academically interesting and fundamental questions.

Computer: So are you saying that new areas of research will come less from trying to fill some identified need and more as a direct result of observing the technology itself?

Wulf: Yes, people talk about "curiosity-driven research"—research motivated solely by the researcher's desire to understand the natural world. I'm sure there's some of that, but an awful lot of research is need-driven, motivated by a problem domain. It is research whose question is triggered by observing a man-made artifact, rather than a natural one. Pure science is motivated by trying to understand the natural world, but as we build more complex man-made systems, we will see behaviors that are every bit as difficult to understand, every bit as fascinating, and every bit as important as understanding the natural world.

Computer: On the other hand, there seems to be a movement away from technology for technology's sake. We've seen an almost frantic movement to have computers conform to the way people behave.

Wulf: It's only because the technology has become so much more versatile that we can even consider making it more humane. I remember back in the '70s, before PCs, when several people made the rather startling but correct observation that it takes more computing power to support a secretary than a computer scientist. We now

have that power, and we are beginning to develop more accurate models of human behavior. So with both these things, we can begin to be more humane. Of course, we're not completely there yet, either in computing power or in our understanding of what it means to be humane.

Computer: Some people claim that this is the age of software, and indeed it seems that software developments and problems are at the forefront. Others claim that silicon, although less visible, drives it all. Do you see any convergences between these two?

Wulf: Well, as I said everything is driven by Moore's law, but I do see hardware and software design converging. In both, the problem is how to manage complexity. The design of a 10-million-device chip has nothing to do with electronics and everything to do with a digital abstraction of those electronics. As design tools like VHDL become more widely used, the hardware and software engineering of these artifacts will blend. You won't know or care whether you are designing hardware or software.

We made an absolutely arbitrary decision about using the instruction set as the boundary between hardware and software. If you go back to microcode days, the instruction set wasn't really hardware; it was a program running on a simpler piece of hardware. The bugs in a Pentium are software bugs. They manifest themselves in hardware, but in fact they are the same kind of bugs we get in software and for the same reason—complexity. So I'll buy that this is the age of computation and communication but dividing that into hardware and software is absolutely arbitrary, and arguing over the boundary is a counterproductive way to spend time.

Computer: Much has been said about the government, industry, academia triangle, not all of it complimentary. How will these roles evolve?

Wulf: I've been at the Academy now for 17 months, and I've been able to examine this question for fields other than computing and electronics. I have decided that our field is actually doing quite well. Somehow among DARPA and NSF, the really good industrial labs like Xerox PARC, IBM TJ Watson and Bell Labs, and the better experimental universities, a tremendous churning has gone on, and the degree of collaboration and cooperation is phenomenal. It's what a lot of other fields would aspire to. I see considerable respect for industry by academia.

How could you be a university person and *not* respect a Butler Lampson [Turing Award winner for work at Xerox PARC, DEC, and Microsoft]. Industry recognizes that what has gone on in universities has been very beneficial. That's not to say perfect. But I think it is worthwhile to step back and say, this has really worked pretty well. For example, the field would not have progressed nearly as fast if Carver Mead hadn't deposited his model of VLSI design, or Xerox hadn't done the Alto, or DARPA and NSF hadn't made it easier for universities to do VLSI design through MOSIS.

Computer: But now funding through services like MOSIS is shrinking. What do you see as an alternative? Do you see industries subcontracting university research? Will the arrangement change from less of an isolated laboratory to more of a collaboration, almost a business partnership?

Wulf: I hope that doesn't happen. Dick Cyert was president of CMU when I was a wet-behind-the-ears assistant professor and he was fond of talking about an organization's unfair advantages—the properties of an organization that made it better at doing what it does than anybody else. His idea was to figure out your unfair advantage and use it, don't try to compete on somebody else's turf when they have the unfair advantage.

Universities it seems, have the unfair advantage of being able to scan the horizon relatively unfettered to see where the interesting problems are. They have a cost advantage for doing some kind of research. Universities have a lot of unfair advantages, but one unfair advantage they absolutely do *not* have is an understanding of customers and business requirements. So if you want to see products get really screwed up, do them in a university. It's just not the right place, and not for some abstract intellectual reason, but simply because universities are not good at it.

Computer: And yet academia would argue that industry has the unfair advantage of dollars, of being able to hire their people away before they are completely educated.

Wulf: That's a bit of a problem, but I think a temporary one. We have a relatively severe shortage of information processing personnel, so people are doing some crazy things. But I have a lot of faith that top-level management won't eat the seed corn.

Computer: So putting your professor hat on for a moment, do you have any thoughts

about how university people can contribute more to the progress of computing?

Wulf: I do. We need to do more to encourage technological literacy across disciplines. We teach physics for poets; why don't we teach engineering for poets as well? My favorite quick definition of engineering is "design under constraints." What we do is design things and we do it under a set of societal, business, and technological constraints. Design is an incredibly creative activity. Why we're not teaching it to liberal arts folks is confounding.

Computer: In another definition of engineering, you stated, "I feel strongly that engineering is an activity that creates artifacts and processes that improve people's lives." How will that charter be fulfilled? Will we see innovation in the next decade?

Wulf: I'd like to first distinguish between invention and innovation. Invention is the act of discovery. Innovation is taking the discovery and producing products that actually sell and affect people's lives. A lot gets invented; a smaller amount gets innovated. Almost nothing that goes on in an industrial laboratory is concerned with invention.

Most people in industrial laboratories are trying to produce an existing product that is faster, cheaper, and more reliable in a more user-friendly way. But they are essentially dealing with an existing market and trying to better satisfy the needs of that market at a lower cost. The true inventions, the breakthrough things, are rare events and they are seldom recognized for what they are at the time they happen. Alexander Graham Bell thought the telephone was a broadcast device. He was going to use it to deliver concerts to people. Marconi thought radio was a point-to-point communication device between two people. When Bell Labs invented the laser, their patent attorneys initially refused to file a patent on it because they could see no implications for communications. So going back to your question, the path for innovating breakthrough technologies is probably not something you can predict—which is why it tends to get done in small companies, rather than large ones.

Computer: So are you saying there is no clear path to innovation?

Wulf: There's a wonderful book from the Harvard Business School, called *The Innovator's Dilemma*. It explains what lots of us have observed about the computer business—things like why the mainframe manufacturers missed the introduction of the minicomputer and the minicomputer

people missed the introduction of the PC. The author, Clayton Christensen, argues that this is so for all truly breakthrough technologies. And it's not because the big company is somehow dumb or unaware; it's that at the time the innovation is developed, it doesn't satisfy the needs of the existing customer base. So when the PC comes in, it doesn't satisfy the needs of the minicomputer users and it gets shelved in favor of efforts to improve the minicomputer. By the time the PC gets to the point where it *can* do what the minicomputer can do, the smaller companies, who weren't hampered by the needs of an existing customer base, have already cornered the market.

Computer: It seems that innovation stems from new and interesting areas in which computing intersects some application. What are the challenges to making existing intersections larger and what new intersections do you see in the near future?

Wulf: If you want the interesting intersections, you have to do a very hard thing: you must examine the unstated assumptions you are making about the way the technology will be used, and these may be so deeply ingrained that you aren't even aware of them.

I experienced this firsthand six or seven years ago. I was on a committee at the University of Virginia tasked with determining how information technology was going to affect the university over the next several decades. This included things like the humanities. Being a standard techie, I assumed word processors might be useful to humanists, but offhand I couldn't think of anything else. On one of those light-bulb-go-off days, I suddenly realized there were applications I had never thought about. An historian and an English professor were on the committee. Eventually the three of us entrepreneured the Institute for Advanced Technology in Humanities, which aims to understand and support the scholarship of humanities with modern technology. In the process, they are redefining the limits of humanistic scholarship.

So what I have learned is that if you question your own unstated assumptions, you will frequently find intersections you never thought of, and in so doing you can fundamentally change a great many areas. This is ultimately what innovation is all about. ❖

William Wulf is president of the National Academy of Engineering and AT&T Professor of Engineering and Applied Sciences at the University of Virginia. Contact him at wwulf@nae.org.