

Vol. 2 No. 6

acm

# QUEUE

tomorrow's computing today

September 2004

\$4.95

Gordon Bell on  
Standards  
Subterfuge

Schizoid Classes

## Riding the VoIP Wave

PBX Wipeout

You Don't Know Jack about VoIP

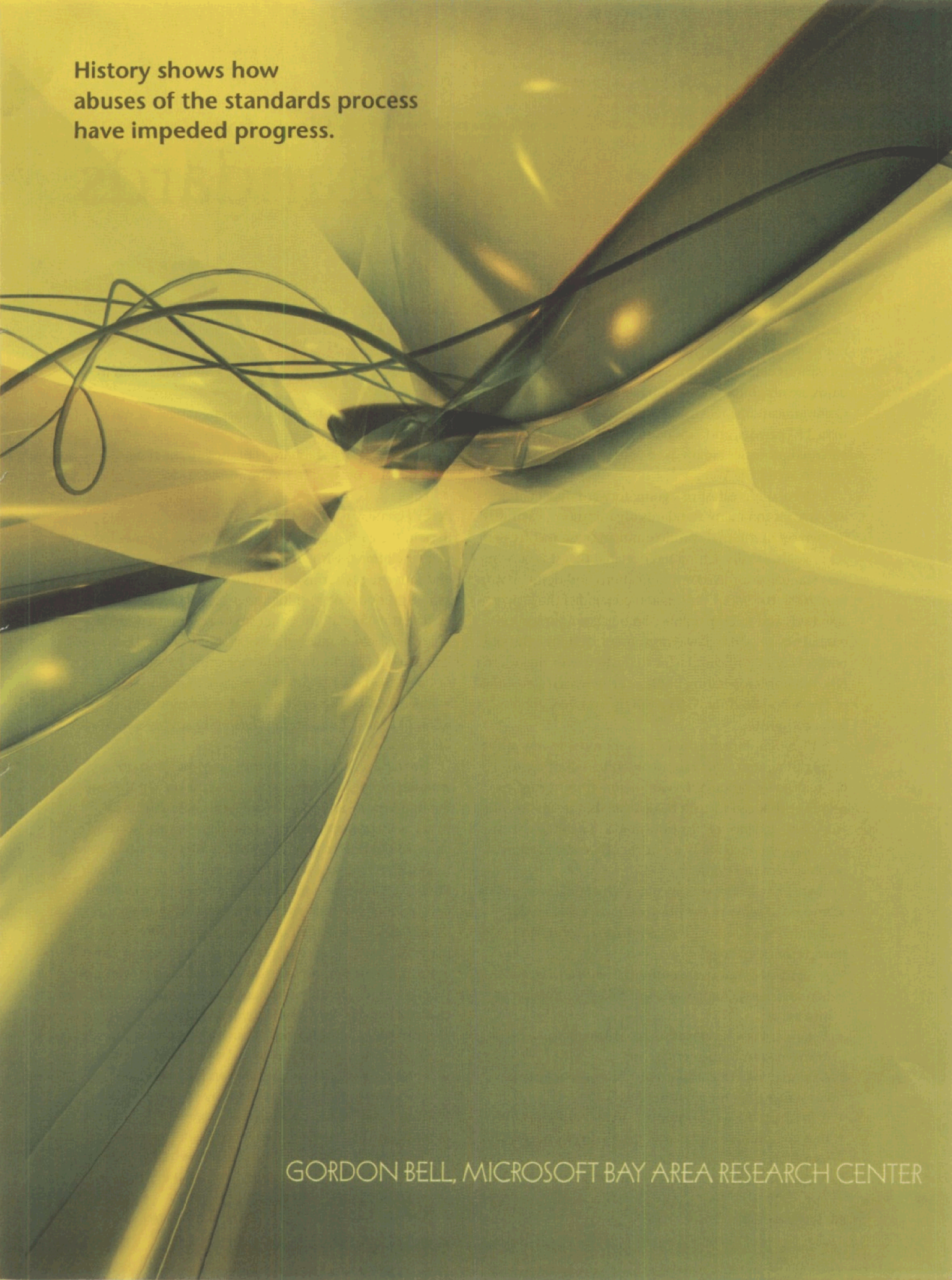
Security Ripple Effects

# A Time and a Place for Standards

Over the next decade, we will encounter at least three major opportunities where success will hinge largely on our ability to define appropriate standards. That's because intelligently crafted standards that surface at just the right time can do much to nurture nascent industries and encourage product development simply by creating a trusted and reliable basis for interoperability. From where I stand, the three specific areas I see as particularly promising are: (1) all telecommunications and computing capabilities that work together to facilitate collaborative work; (2) hybrid computing/home entertainment products providing for the online distribution of audio and/or video content; and (3) wireless sensor and network platforms (the sort that some hope the 802.15.4 and ZigBee Alliance standards will ultimately enable). No doubt there will be others, but for the purposes of this discussion, these should suffice.

The point here is that, in each of these areas, the right standards adopted at the right time can make an important contribution to technical evolution by applying critical design constraints. That is, they can do much to conserve vital design time and effort simply by providing a stable foundation of already defined compute capabilities and processes. Thus, instead of starting each new system from silicon, developers can be liberated to turn their attention to the design of higher-level, value-added functionality.

This is not to suggest that standards offer any sort of panacea. Far from it. In fact, much of this article is devoted to considering the foibles and all-too-familiar



History shows how  
abuses of the standards process  
have impeded progress.

GORDON BELL, MICROSOFT BAY AREA RESEARCH CENTER

# A Time and a Place for Standards

failings common to all SSOs (standards-setting organizations), including ITU, ETSI, IEEE, ATIS T1, TIA, and IETF, not to mention the myriad of consortia created to push for various communications standards. The fact is, standards are ubiquitous. GSM (Global Systems for Mobile Communications) telephones, for example, rest on more than 150 patents, and Tim Berners-Lee estimates that 30 patented processes are invoked with each mouse click used to access the Web.<sup>1</sup>

Indeed, our greatest risk going forward may be that we have far too many standards organizations, each with its own set of internal conflicts and an often inconsistent set of goals. Finally, China has declared that it is creating new standards for telecommunications and home A/V. In May 2004, the 9/11 Commission concluded the lack of standards and interoperability among communications, emergency response, law enforcement, and firefighting bodies was a significant factor in dealing with the disaster. Like other interoperating bodies, the newly created Office of Homeland Security is overseen by over 90 Congressional committees.

Still, when implemented in an appropriate way at the proper time, standards can do much to reduce wasteful, redundant product development—thus freeing up resources that can instead be dedicated to fresh, inventive work. For evidence of just how much of a difference this can make, you need look no further than the great PC revolution of the 1980s.

Rob Gingell, Sun Microsystems' chief engineer, provides a good sense of the positive impact well-crafted, well-timed standards can have with these observations on the economic ecology:<sup>2</sup>

1. A standard delineates a point of homogeneity, enabling heterogeneity, change, and unbridled innovation in other areas.
2. A standard is a specification to which an artifact conforms, not an implementation.
3. A standard is more important for how it affects the consumer than for what it offers.
4. A standard has a community—apply it only to affect that community, and expect it only to affect that community.
5. A standard is as strong as its enforcement mechanism,

though this varies with time.

6. Consumer investments are never to be undone by a standard.
7. Innovation to the standard must come with “skin in the game.”
8. Innovation must be “within chaotic range” of the standard.
9. The lifetime of a standard is limited to the time it enables innovation in its connected areas.

## A FEW GENERAL PRINCIPLES

Back in the early 1980s, I looked out over a fragmented market cluttered with all manner of proprietary hardware and software platforms created by a multitude of organizations in hopes of defining the personal computer, the workstation, and all the other small systems that made use of a variety of microprocessors, bus interconnects, and dialects of Unix and other operating systems. It was not a particularly reassuring view. So I found myself wondering: Was this disconnected, overlapping mess the result of a lack of industry leadership or simply a shortage of capable engineers?

There was good reason in those days to question whether our field was ever going to grow up. On the one hand, hardware and software vendors alike continued to act as if the mainframe era had never ended—as they churned out one proprietary product after another in hopes of locking in a second generation of unwary customers. Meanwhile, some vendors were doing their very best to influence, redirect, or subvert existing standards so as to better serve their own proprietary interests. One particularly irritating example came in the form of the many different proprietary extensions created by vendors for languages such as Fortran and SQL—virtually all of these extensions, of course, having been designed with an eye to carving out monopoly franchises.

To some degree, this is only to be expected. Companies and even countries, after all, can always be counted on to seek out their best advantage by whatever means. Conventional business school logic argues that standard products are bad, since the only way to differentiate one from another is by price—which inevitably leads to ruinous price-cutting. Although I don't believe that is

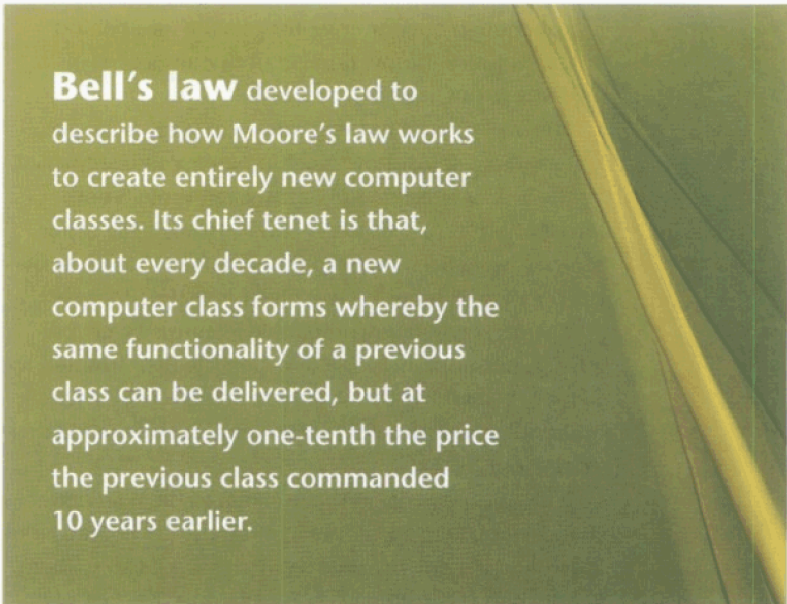
necessarily true in an industry such as ours where product life expectancies are exceptionally brief, most MBA-bearing executives certainly *believed* it to be true in the early 1980s. But they were also painfully aware of all those bothersome customers that kept insisting on the adoption and support of industry standards. The obvious solution for some vendors was to throw their support behind faux standards, containing cloaked, yet potent, proprietary advantages designed to serve their own interests. This was particularly evident in the Unix community, where all manner of company-specific dialects sprouted up like so many weeds. Owing either to arrogance or innocence (or possibly both), the vendors that played the faux standards game all made the fatal assumption that most of the buyers could be fooled most of the time.

Of course, these vendors succeeded only in deluding themselves. That's because it's been shown time and again that genuine standards created to serve the true interests of consumers will also generally end up serving the best interests of vendors. Many examples reaffirm this principle, and I'll continue to cite them as this article unfolds. But let's pause briefly here to consider faux standards such as SQL and Unix/Posix that failed to provide for the testing of interoperability and thus became little more than meaningless check-box requirements. Now compare that with serious networking protocols such as TCP/IP, HTTP/HTML, XML, and SOAP, each of which demands real interoperability across a wide variety of different products (since, otherwise, nothing would work). Note also that with each of these networking examples, the customer's need for a commodity standard (something that can effectively guarantee at least two sources for each required product), interoperability, and stability have been ably satisfied through the introduction of reasonable standards. It turns out that vendor interests have also subsequently benefited as the market in each of these areas has experienced substantial growth.

Further note that each of the standards indicated here—faux, legitimate, or otherwise—is limited to some horizontal level of integration common to many different types of applications and product-segmented industries. This wasn't true of the earliest de facto, vendor-dominated computing standards, which were characterized instead by complete vertical integration. That is, vendors went to market with systems that consisted entirely of proprietary circuits, system designs, peripherals, operating systems, languages, applications, and content. It wasn't until languages such as Fortran and Cobol surfaced that multivendor standards started to emerge.

Each of the successful standards that has followed has similarly been limited to just one stratum. For example, SQL was intended to operate strictly at the middleware level; Unix/Posix was an operating system standard designed explicitly to be both application- and platform-agnostic (although, ironically, the need for interoperability between different flavors of Unix was blithely overlooked); and TCP/IP, HTTP/HTML, XML, and SOAP have been intended all along to work only at the network protocol level.

There's also a Darwinian dimension to all this since, in some respect, it could be said that in the first three decades of commercial computing, DEC, Hewlett Packard, IBM, and Sun used the vertical-integration approach to artificially bolster the sales of inherently weak products (disks, for example). You could argue that they managed to hide some second-rate, high-margin products in the stack—and those, by and large, were products that probably could never have survived on their own. In a world dominated by horizontal standards, though, weak



**Bell's law** developed to describe how Moore's law works to create entirely new computer classes. Its chief tenet is that, about every decade, a new computer class forms whereby the same functionality of a previous class can be delivered, but at approximately one-tenth the price the previous class commanded 10 years earlier.

products have fewer places to hide. Generally, they live or die on their own merits, with only the strongest surviving long enough to spawn offspring. According to the Darwinian view, this is an effective way of ensuring that only the best technological genes are propagated.

There's something else the evolution of computing standards has in common with the evolution of species: the fact that timing is everything. Even the cleanest specification pitched at an important level of integration won't have much of an impact unless it surfaces at just

# A Time and a Place for Standards

the right time. We can thank MIT's Dave Clark, one of the people who helped define the Internet and continues to lead standards efforts as a member of the IETF (Internet Engineering Task Force), for an eloquent explanation of why this is so:

Standards setting sits in a boring trough between two exciting peaks. The first is the peak of technical innovation, and the second is the billion-dollar investment *or market*. The problem is that there is not enough space *or time* in the trough to get the job done. As soon as someone sees the opportunity for a billion-dollar investment, they trample you from behind. They complain that standards setting takes too long, but there is no way it could happen fast enough once an investor gets the idea behind the technical innovation. So in that respect, standards setting is a thankless task.<sup>3</sup>

The evolution of browser interfaces gives us an excellent example of just how much progress can be impeded by the lack of effective standards. Even though both Microsoft's Internet Explorer and Netscape's Navigator browsers were licensed from the same basic source at the University of Illinois, each evolved over time to incorporate many wholly incompatible features. As a consequence, both products had to be repeatedly retrofitted to adapt to something the other one offered—an exercise in futility that was wasteful for the vendors and maddening for the users.

It also bears mention that a standard has a far better chance of making a real impact if no royalty is charged to those who employ it. You'd think this would go without saying, but, sadly, it doesn't. For example, the fact that Xerox was willing to provide a royalty-free license for its Ethernet technology proved to be a significant factor contributing to the general adoption of 802.11. In contrast, IBM paid an inventor for the Token Ring patent, and ultimately that royalty worked to erode support for the ring's adoption.

## RULES OF THE GAME

The last two decades have provided us with plenty of lessons regarding how to go about setting standards—as well as how *not* to do it. In the paragraphs that follow, I relate

some of the lessons I've learned along the way. Please note that these "rules" are by no means complete—nor even necessarily consistent—but they do fairly represent the observations I've been able to make over a lifetime of deep involvement in the game.

**You get no product, technology, or standard before its time.** The market's readiness for a given standard and the wherewithal to actually implement the standard depend on both the state of technical development in the area of interest *and* the availability of sufficient processing speed, memory, storage size, and network bandwidth to support the capabilities in question. Although it's safe to assume rapid increases in computing power and capacity, some proponents—in their enthusiasm to promote one technology or another—have gone so far as to assume advances beyond those suggested by Moore's law. This is almost never a good idea. Thus, for example, people pushing for object-oriented programming standards in 1990 were disappointed to discover that they were somewhat ahead of the curve for the underlying technology.

Conversely, Microsoft decided in that same year to scuttle its program to develop pen-based computing capabilities, not long after the pen-based computing startup GO had decided to close its doors. Microsoft took quite a lot of flak for that, but notice that Palm's first pen-based system didn't come out until some years later—and even then only because the company decided to live with some rather severe compromises in the user interface. It wasn't until 2002 that Microsoft finally released its first tablet computer capable of recognizing a range of cursive handwriting, prompting industry observer Stewart Alsop to comment, "If GO had ever been given the chance, they would have succeeded."<sup>4</sup> It's worth noting, however, that Microsoft's tablet computers packed more than 50 times as much processing power and 100 times as much memory as was available with the earliest GO systems.

My point here is simply that a truly useful pen-based computer was not really feasible in 1990, given the processing speeds, memory sizes, and algorithms available to system designers of the day—a point that even the *New York Times* technology reporter John Markoff missed when he later revisited the history of pen computing.<sup>5</sup> A similar scenario confronts today's promoters of RFID

(radio frequency identification) tag technology, who optimistically look past the cost and reliability problems that stubbornly refuse to go away. Mind you, I believe those problems will ultimately be overcome—but not until that technology's time has come (and I don't believe that time is quite yet).

**Ultimately, each high-volume information technology product becomes a commodity distributed free of charge or given away with something else.** When a new technology—a video encoder, for example—is introduced, it often will come in the form of after-market, add-on hardware aimed at a few specialized, high-end markets. As system power and capacity grows in accordance with Moore's law, however, software implementations of those same capabilities will generally get rolled into new systems at no apparent extra charge. A familiar example is offered by all the new capabilities for processing video and audio streams that have been introduced over the years. You've probably noticed yourself just how short the shelf-life of those add-on cards tends to be. Well, there's a law that explains that: Bell's law, developed to describe how Moore's law works to create entirely new computer classes. Its chief tenet is that, about every decade, a new computer class forms whereby the same functionality of a previous class can be delivered, but at approximately one-tenth the price the previous class commanded 10 years earlier. The new offering typically will include a new platform, a new network, and a new interface that allows new applications.

**Once a technology has been proven in a hardware implementation, it proceeds to be implemented in software running on high-performance processors,** which in turn continues to evolve at an even faster rate until it can finally be implemented as a zero-cost option on small areas of silicon and iron oxide. Is there no end to the bad news for patent holders of algorithms (for example, a codec)? Once a capability has been implemented as software, it essentially slips from their grasp—both technically and in terms of their ability to extract a revenue stream. That's because the technology starts evolving from that point forward at the rate described in Moore's law for software. In many instances, in fact, the rate of software progress has actually exceeded Moore's law—at the rate of 60 percent per year for some programs and algorithms.

One reason for this brisk rate of change is that the software development market offers exceedingly low barriers to entry. That is, since it doesn't take much capital to start new operations, you get plenty of players and thus some extremely vigorous competition. But in this dog-eat-dog

world, the question arises: How do you best manage this brutal evolutionary process for the benefit of all? And this, of course, brings us back to the central premise of this article: standards still have a very important role to play, as with the plethora of audio and video codecs.

**People buy products, not protocols or algorithms.** A little more than a decade ago, it looked as if the people who developed MPEG coding would make a bundle since, at the time, encoding and decoding files required a special off-board processor. Before long, however, CPUs powerful enough to handle the MPEG processing on their own came to market, suddenly extinguishing any demand for special MPEG chips. Because this also had the effect of eliminating any place for patent royalties to “hide,” fantasies of fortunes to be derived from MPEG were quickly dispelled.

Similarly, you have to wonder whether an organization such as Real Audio will be able to maintain any royalty stream since wire protocols have proven to be notoriously poor places to collect royalties. The Dolby patents, on the other hand, perhaps present us with an exception—or possibly a proof by way of exception—since the Dolby methods for processing sound have spanned the gamut from reel-to-reel tape to theatre sound systems—generating a strong IP revenue stream at each point along the technology curve. Any hopes for future Dolby-like success stories, though, dim in light of the vast number of overlapping patents and standards-setting organizations that currently compete to regulate IP. Frankly, I see this as perhaps our greatest current and future threat to innovation. The SSOs represent a double whammy, as several organizations and even countries can—and often do—claim jurisdiction over a new invention or standard and then proceed to set conflicting goals and constraints.

**Either make the standard or follow the standard.** To understand and appreciate the reason for this rule, you need look no further than the PC revolution. After the Intel x86 architecture initially emerged as a standard at the processor level, standards at the peripherals, system packaging, and operating system levels emerged over the next few years and continued to evolve. What came of all that was an industry that took off with tremendous propulsion. Contrary to the common wisdom of the early 1980s, which argued that any standardization would tend only to inhibit innovation, the PC phenomenon provided for tremendous innovation at each horizontal layer in the architecture. In contrast to the classic vertically integrated architectures of the preceding three decades, where a single manufacturer defined everything from the

# A Time and a Place for Standards

silicon level up to the applications and content levels, the PC architecture simply specified the interfaces between each of the different layers. This had the advantage of ensuring a large overall market, while also leaving considerable room for innovation and competition at each level (chip, operating system, language, network, applications, and content).

Meanwhile, hardware manufacturers such as Hewlett-Packard, IBM, and Sun continued to follow a vertical strategy where their mid-range systems were concerned. Although they did quite well, we can say in retrospect that they didn't fare nearly as well as did their counterparts in the PC world. That's because they ended up on the wrong side of the economies-of-scale equation and so were faced with costs as much as 10 times greater than those in the PC world, and they also ended up realizing performance gains over time that paled in comparison with those achieved in the PC camp.

**If you set out to create your own standard, make sure it's one that will stick.** Why not just go for it? Because when you try to set the standard yourself but fail, you get an opportunity to do *two* implementations instead of one. Still, either way, at the end of the day you've got to throw all your weight behind the standard that ultimately prevails. The direction of the x86 architecture, with its current extensions to 64 bits, is a case in point—with the end yet to be determined. Almost a decade ago, Intel began working with HP to create a new architecture to extend the x86 for 64-bit addresses. But that platform proved to be so complex, so difficult to compile for, and so burdened with patents that no one could technically, financially, or legally afford to create an alternative implementation.

In the current instance, AMD has created a 64-bit, upward-compatible, evolutionary extension that builds on the x86 legacy. So Intel has been forced to backtrack and create an AMD-compatible processor. The next chapter of the story will likely come one or two generations from now with Intel providing very complex chips that implement both processors, while AMD provides dies that produce many more of their simple processors.

**Be prepared to react and follow once the standard changes.** Over the course of the PC revolution, IBM had to learn this lesson the hard way. First, once the PC

really started to take off, Big Blue tried to take control by creating a new bus for the PC architecture. This succeeded only in driving business away as customers made it clear they'd really prefer to stick with the de facto—and wholly *license-free*—standard bus architecture. IBM grudgingly was forced to follow along. Not to be deterred indefinitely, however, the company later tried much the same gambit up at the operating system level with an attempt to displace DOS with OS/2. Once again, the battle was lost—at a tremendous cost in market share in a market that was growing more lucrative by the day. IBM seems to have finally learned its lesson since it now is out in front of the effort to make Linux a common operating system standard—and it would appear this is working out nicely for IBM, as well as for the Linux community as a whole.

But there's also another possible strategy. Apple, for example, is to be congratulated for having *not* followed the PC standard. That's because it wanted to have complete control over its own architecture—which indeed over the years has continued to offer a significant and meaningful alternative to the PC.

The point here is that deviations from the norm are essential for the sake of progress. That said, those organizations that choose to embark on this path must be cognizant of the risks, since spectacular failure is not only a possibility but also a likelihood. Note also that “meaningful deviations” does not refer just to the mere repackaging of old ideas or to the creation of alternative approaches for simple functions. A meaningful deviation is something that's both sweeping in scope and well considered—ultimately yielding some new capability that branches off from the ordinary to create an entirely new evolutionary path.

**Always bear in mind that “better” is the enemy of “good enough.”** The way in which the Ethernet standard has evolved over the years is extremely instructive. The initial physical layer and protocol definition came together quite quickly in response to a clear and present market requirement. Xerox and DEC were looking to employ Ethernet as the backbone of their respective product strategies—and Intel was looking forward to selling the chips. Moreover, the initial design was able to



draw heavily on experience gained with Xerox PARC's early LAN (local area network), and so could be developed expeditiously by just eight engineers from the three companies. The mere fact that a standard could be created with that much market clout right from the start was a monumental achievement in its own right.

By comparison, the actual implementation details related to modulation (broadband versus baseband) and topology (buses, rings, trees, or centralized switches) were of small consequence—chiefly because any cost or performance gains to be realized through optimization would obviously be dwarfed by what was to be gained by having a single standard to rally around. Nevertheless, variations on the earliest 802 series Ethernet definitions were legion, as one company after another came up with its own flavor in an effort to put its imprimatur on the standard or to use it as a point of market differentiation. All of this served only to confuse the market and delay work on the development of *real* Ethernet systems by at least five years—at an untold cost in wasted research money, lost productivity, and needless redundancy. Also, work on a 100-Mbps (megabit-per-second) version of the Ethernet standard was set back by at least five years while silly battles continued to be waged over the 10-Mbps standard.

**Less truly is more.** It may be that “more is merrier,” but in the standards realm, many merry folks almost never make for a happy result. In keeping with Fred Brooks' rule of thumb, I believe the number of entities involved in any design process should be kept small—ideally, at least three, but never more than seven. That's why I cringe at the very mention of using “consortia” or “alliances” to define standards. Frankly, these efforts are generally doomed right from the start for a variety of reasons:

- The best standards draw on a relatively narrow understanding of technology to address a fairly small problem set. As more and more players are invited to the party, things unravel as the effort tends to expand both in terms of design goals *and* constraints—leading at first to thrashing and ultimately to paralysis. Fortran offers an illuminating example here since a “big tent” approach to defining a standard version succeeded only in side-tracking the language's evolution for *decades*.
- With consortia and alliances, there's a disturbing tendency toward standards overlap. For example, the people working on a low-power, low-speed Bluetooth standard lost sight of the fact that their goal was to provide for wireless interconnect capabilities over a few meters. Thus, they ended up also trying to address the higher-power, higher-speed 802.11b challenge, intended to provide for interoperation at distances up

to 50 meters. The results of this muddled focus proved predictably disastrous.

- Unless each participant in the standards process absolutely depends on the process succeeding, it most likely will fail. As evidence, consider just how many dilettantes managed to insinuate themselves into all the various Unix standards efforts. And look at what came out of that: total fragmentation. But, on a positive note, it did give rise to the canonical joke about standards: “The nicest thing about Unix standards is that there are so many to choose from.”

**Standards should be based on real experience, not on committee designs.** Perhaps an even better way of putting this would be: “If you haven't actually lived with the design proposed as a standard, don't adopt it.” The best way to establish whether a specification is real or not is to implement several alternative interfaces. In fact, the IETF has set just such a rule for itself, holding that no standard can be created unless there already are at least two interoperating implementations. Similarly, computer users who hope to use a particular standard to leverage their buying power should always take care to test their systems on two separate implementations before deciding whether to link their fates to that standard. The Grid community in particular would be well advised to adopt just such a discipline before wedding itself to standards that define its future. Unfortunately, the Grid software is being done in a monopolistic fashion by a few government labs and not in the fashion of IETF.

**The fewer standards, the better.** A common aim for all standards should be to unify a whole *set* of variables. In my book, a “set” should ideally consist of more than two parts. The idea, in other words, is to leave as little room as possible for other standards, because having too many standards is like not having any standards at all. Recall, for example, the absolute torrent of 802 LAN standards that issued forth during the early 1980s. It seemed that every company, consortium, or committee had its own particular flavor of the month. In the original paper defining the Ethernet spec, I argued for LAN standard birth control to prevent just such madness. Ultimately, of course, we ended up with only one Ethernet LAN standard.

Things proved much tidier over on the bus front, largely because of the sheer complexity of the chip/board interface and the fact that Intel had an interconnect standard it wanted to push. Now we find ourselves faced with a plethora of mobile standards and protocols, all of which are tainted with built-in proprietary advantages for one vendor or another. The ultimate effect of this will simply be to push back the time when it will truly be pos-

# A Time and a Place for Standards

sible to migrate to a mobile Web, which is unfortunate for vendors and users alike.

## SETTING STANDARDS FOR THE DECADE AHEAD

I started this article by outlining several areas of burgeoning technological innovation that appear to be ripe for new and evolving standards. But, as I hope I've managed to illustrate, those areas need to be approached with some restraint since too many standards are at least as bad as none at all. Hence, to guard against that danger, I think the time has come to declare a moratorium on creating even more consortia and SSOs than we already have. I also think it's high time that we insist that no standard be exposed to double jeopardy. That is, one committee per standard should suffice. Period.

That said, the convergence of traditional telecommunications and the Web already is well under way, yet we have no standards for IP telephony on which to build the collaboration tools that people are clamoring for. Similarly, standards for messaging, video telephony, and video collaboration are presently notable by their absence.

Anyone looking to develop products that bridge the computing and home entertainment realms, meanwhile, face a hodge-podge of requirements: the audio/video standards that apply in the consumer electronics industry, the digital rights management demands of the entertainment industry, and the communication standards that have been adopted in the computing industry—most particularly USB (universal serial bus) and 802.11x. Somehow, all this complexity must be controlled so all developers and manufacturers can finally have a chance to read from the same sheet of music.

Finally, in the wireless sensor network area, while various IEEE standards are in process and several early products are just starting to emerge, nothing yet delivers the sort of performance and reliability capable of spawning a robust industry. This has created an aura of uncertainty, which in all likelihood will abide until sufficiently robust technology begins to emerge. Then, and only then, can standardization begin in earnest. To push those efforts prematurely is to hope vainly that standards and consortia can somehow drive the technology to maturity. But those of us who have been around long enough to see

history repeat itself a few times can attest that it actually must be the technology that drives the standard. It just doesn't work any other way.

## ACKNOWLEDGMENTS

This article is dedicated to all the engineers who devote their energy to working on standards so as to create a more orderly environment for the benefit of all. ☐

## REFERENCES

1. Cargill, C.F. The Sisyphus Agenda: Standardization as a guardian of innovation. In *The Standards Edge: Dynamic Tension*, ed. S. Bolin, 31-46. Sheridan Books, 2004.
2. Gingell, R. Standards as economic ecology: A system in tension. In *The Standards Edge: Dynamic Tension*, ed. S. Bolin, 7-14. Sheridan Books, 2004.
3. In this passage, excerpted from a recent e-mail (February 23, 2004), MIT's Dave Clark recalls a presentation about standards setting that he made approximately 15 to 20 years ago.
4. Markoff, J. Silicon Valley's dream tablet, from Microsoft. *New York Times*, Nov. 6, 2002.
5. Markoff, J. Newly released documents shed light on Microsoft tactics. *New York Times*, Mar. 24, 2004.

## LOVE IT, HATE IT? LET US KNOW

feedback@acmqueue.com or [www.acmqueue.com/forums](http://www.acmqueue.com/forums)

**GORDON BELL** spent 23 years (1960-1983) at DEC (Digital Equipment Corporation) as vice president of research and development. He was the architect of various mini-computers and time-sharing systems (including the PDP-6) and led the development of DEC's VAX and VAX Computing Environment. Altogether, Bell has been involved in—or responsible for—the design of many products at Digital, Encore, Ardent, and a score of other companies. Currently, he is a senior researcher in Microsoft's Media Presence Research Group, located in the company's Bay Area Research Center (BARC). He previously wrote about this topic 20 years ago: Bell, C. G. Standards can help us. *Computer* 17, 6 (June 1984), 71-78; <http://research.microsoft.com/users/gBell/gbvita.htm>.

© 2004 ACM 1542-7730/04/0600 \$5.00