Disappearing Hardware

In Mark Weiser's vision of ubiquitous computing, computers disappear from conscious thought. From a hardware perspective, the authors examine how far we've succeeded in implementing this vision and how far we have to go.

Roy Want and Trevor Pering
Intel Research, Santa Clara

Gaetano Borriello
University of Washington and Intel Research, Seattle

Keith I. Farkas
Compaq Western Research Laboratory

For many tasks today, the use of computers is not entirely satisfactory. The interactions take effort and are often difficult. The traditional, and still prevalent, computing experience is sitting in front of a box, our attention completely absorbed in the dialog required to complete the details of a greater task. Putting this in perspective, the real objective is the task's completion, not the interaction with the tools we use to perform it.

To illustrate the point further, if somebody asks you for an electric drill, do they want to use a drill, or do they really want a hole? The answer is probably the latter—but computers are currently very much a drill, requiring knowledge, training, effort, and skill to use correctly. Creating a hole is relatively simple, and many hidden computers invisibly accomplish what seem to be simple tasks, such as regulating our cars' brakes. But unlike other inanimate objects, a computer system might be able to infer the result autonomously and affect the desired outcome, increasing both its potential and end-user complexity. Realizing this potential while managing the complexity is the fundamental challenge facing computer system researchers.

An important trend over the last decade is the emergence of specialized, task-specific hardware (such as spell checkers, calculators, electronic translators, electronic books, and Web pads). These devices have a specialized interface and address the desired goal of ease of use. In contrast, a PC is a generalized machine, which makes it attractive to purchase—the one-time investment having the potential for many different uses. However, in other ways it adds a level of complexity and formalism that hinders the casual user.

Mark Weiser wanted to explore whether we could design radically new kinds of computer systems. These systems would allow the orchestration of devices with nontraditional form factors that lend themselves to more natural, tacit interaction. They would take into account the space in which people worked, allowing positional and manipulative[1]—rather than just keyboard and mouse—interactions. Along with specialization and the use of embedded computers, support for mobile computing and wireless data networks is an important facet of this vision—in other words, invisible connectivity. A goal of this exploration is that we would learn to build computer systems that do not distract the user; ideally, the user might even forget the hardware is present.[2] In essence, Weiser was proposing that well-designed computer systems would become invisible to the user and that our conscious notion of computer hardware would begin to disappear. Some years later, Don Norman popularized this concept in his book *The Invisible Computer*.[3]

In this article, we survey the progress toward Weiser's vision from a hardware viewpoint. Where

1536-1268/02/$17.00 © 2002 IEEE

Figure 1. Hardware improvement over the last decade: (a) the Xerox ParcTab, the first context-sensitive computer (1992). The design shows the limited display available at that time—a 128- × 64-pixel, monochrome LCD. (b) a typical PDA available today with a color 240- × 320-pixel VGA (transreflective) screen.

have we been, where are we, and where are we headed? What characteristics will make hardware disappear from our consciousness, and what will it take to achieve them?

Where we've been

The research community embraced Weiser's call to explore ubiquitous computing. For example, his vision inspired the work at the Xerox Palo Alto Research Center (PARC) in the early 1990s and such projects as ParcTab,[4] Mpad,[5] and Liveboard.[6] Olivetti Research's Active Badge[7] and Berkeley's InfoPad[8] projects also embraced this research direction, as did other notable centers of excellence, such as at Carnegie Mellon University, IBM, Rutgers University, Georgia Tech, and the University of Washington. Unfortunately, many of the early systems were based on technologies that were barely adequate for the task, so they fell short of designer expectations.

Figures 1a and 1b illustrate the extent of hardware improvement over the last decade. In 1990, no Wireless Local Area Network standards existed; the processors suitable for mobile devices operated at only a few megahertz, while PCs were typically shipping with up to 50-MHz processors. The early electronic organizers (pen-based PDAs had not been invented) proudly claimed 128 Kbytes of memory, while PCs shipped with 30-Mbyte disks. The displays were also quite crude: laptops used monochrome VGA, and the few handheld devices available mainly used character-based displays.

Industry soon responded to the challenge with a tighter focus on mobile computing. A flurry of early products hit the market, particularly in the tablet style, that tried to make using computers feel more like using a pen and paper. Most of these products have fallen by the wayside: Momenta and EO, IBM's early ThinkPad, and later the Apple Newton, the Casio Zoomer, and General Magic's pad. For these designs, the benefit-to-cost ratio was just not large enough. To be successful, these new devices had to either be better than the traditional pencil-and-paper technology they were replacing or provide desired new functionality. The physical hardware was the dominating factor, and almost every design aspect affected acceptance: size, weight, power consumption, computation speed, richness of interface, and simplicity of design.

We started to cross the acceptability threshold only in the latter half of the decade with the Palm Pilot. It was smaller and lighter, focused on simple applications, and incorporated a novel one-button approach to data synchronization. Finally, an electronic organizer was useful for a significant number of people and had real advantages over the more traditional paper products such as Day-Timers. The computer industry was beginning to move in the right direction.

Where we are

For hardware to disappear from our consciousness, we require transparency of use: if we notice it's there, it's distracting us from our real task. For example, if we notice that we are using a slow wireless network connection instead of just editing our files, then the action of accessing the files is getting in the way of the real task, which is contained in the files themselves. If the link is fast and robust, we will not notice it and can focus on the content. Likewise, if a display can present only a poor representation of a high-quality underlying image, we see a bad display. A high-quality display suspends our belief that the image is only a representation.

The four most notable improvements in hardware technology during the last decade that directly affected ubiquitous computing are wireless networking, processing capability, storage capacity, and high-quality displays. Furthermore, the current popular adoption of emerging technology, such as cell phones and PDAs, strongly indicates that the market is generally ready for advanced new technology. This adoption, however, requires common standards across many products and locales.

Wireless networking

Although progress in wireless connectivity was initially slow, it has increased. This area has witnessed two distinct development trends. The first is in short-range connectivity standards, such as Bluetooth (IEEE 802.15) and the IrDA (Infrared Data Association) standards, which are primarily for simple device-to-device communication. Bluetooth, which will get its first real test in the marketplace in 2002, was designed as a short-range cable replacement, allowing for proximate interaction and the discovery of resources in the user's locality. IrDA had a similar aim. But because infrared signaling requires a line of sight, users had to physically place devices next to each other, often an inconvenience. This technology, which predates Bluetooth by many years, has been considered a market failure. (The sidebar lists URLs for Bluetooth, IrDA, and other areas of interest in this article.)

A truly ubiquitous computing experience requires high-quality displays to let us see through the display process and effortlessly acquire the underlying information.

The second trend is in wireless LAN technology, such as the 11-Mbit-per-second IEEE 802.11b standard and the more recent 54-Mbps IEEE 802.11a standard.

Wireless technologies provide for two basic needs: the ability to detect location and the more basic ability to communicate. In many cases, the ubiquitous computing vision can to some degree be implemented by interpreting simple context information, such as a user's location.[9] Accurately determining in-building locations is difficult. The Global Positioning System (GPS), for the most part, can locate an object only to within 10 meters and does not work in buildings. Short-range wireless standards such as Bluetooth let us more easily discern location and context within a limited area without having to support sophisticated wide-area location technologies. Furthermore, short range implies lower power and the ability to build a smaller device from a more compact energy source, making it more attractive for many human-centric applications.

The emergence of the latter IEEE wireless standards allows for communication cells that span many hundreds of feet with sufficient bandwidth to make us feel as if we were connected to a wired LAN, but without a physical connection's constraints. IEEE 802.11b has already been widely adopted, and 802.11a is expected to follow with higher bandwidths. Next-generation digital cellular networks such as 2.5G (for example, General Packet Radio Service and NTT's DoCoMo—with greater than 24 million users) and the coming 3G networks will extend these capabilities to cover entire metropolitan areas.

The wireless networking of today and the immediate future thus enables portable ubiquitous hardware that remains connected to the global infrastructure. However, to date, wireless networks have lagged behind the bandwidth capabilities of the equivalent wired networks, leaving both the opportunity and the user desire for improved wireless hardware.

Processing capability

For 30 years, processing capability has basically followed Moore's law, which can be summarized as "The number of active devices we can place on a given area of silicon doubles every 18 months"[10] (This is actually a revision of Gordon Moore's 1965 estimate of doubling per year and the later 1995 estimate of doubling every two years.) This trend's obvious consequence is that we can continue to increase the capability of devices fabricated in a given area of silicon. Instead of designing systems built from separate board-level components, we can integrate diverse functionality onto a single chip, resulting in remarkably compact consumer electronics. Similarly, the reduced capacitance resulting from smaller transistor dimensions means that we can operate these devices at higher speeds, increasing their effective performance. Additionally, reduced transistor sizes decrease power consumption, alleviating some of the perpetual problems surrounding energy storage technologies.

The combination of more transistors on a given area of silicon and a reduced power budget has brought us the capabilities of mid-1980's desktop computers in today's battery-operated, handheld PDAs. Two examples are the Motorola Dragonball and Intel StrongARM processors, the most common processors used by today's PDAs. Besides providing low power consumption and high performance, these processors integrate their DRAM and LCD controllers and a host of other interface I/O capability on the same die.

These trends directly affect ubiquitous computing for mobile devices in two ways. First, we can better match algorithmic complexity and execution speed to real-world problems. Second, the resulting power consumption allows for a reasonable operating time before batteries fail. These properties have let us build ubiquitous computing hardware that we can adapt to a greater range of task-specific activities. Nevertheless, further improvement is still required to satisfy the complete ubiquitous computing vision.

Storage capacity

A less visible industry trend is the rate at which storage capacity is improving for rotating magnetic storage and solid-state devices. For 25 years, the capacity of rotating disks has been roughly doubling every year, a rate of improvement faster than Moore's Law! The current storage density found on a disk drive is approximately 10 Gbits per square inch.[11] Today, IBM markets the Microdrive, a one-square-inch device with 1 Gbyte of storage in a compact flash-card format. If this trend continues, in another decade we will be able to carry 1 Tbyte of data in a similar form factor.

From a ubiquitous computing viewpoint, storage is becoming so inexpensive and plentiful that, for many devices, internal storage capacity is not a limiting factor for basic operation. Moreover, we can begin to use storage in extravagant ways by prefetching, caching, and archiving data that might be useful later, lessening the need for continuous network connectivity. Flash memory, DRAM, and Static RAM have all benefited from progress in integration, making large capacities at low prices possible. A typical DRAM chip has approximately 32 Mbytes of capacity; flash memory has 16 Mbytes. Top-end CompactFlash cards, with 512 Mbytes of capacity, closely rival the IBM Microdrive's capacity.

Storage is fundamental to most ubiquitous storage applications. Storage limitations for common tasks hinder the ubiquitous computing experience, forcing us to leave behind information and to carefully select what must be mobile. Abundant storage negates these issues, letting us focus on the important underlying tasks.

High-quality displays

Because vision is one of our most important and acute senses, the need to render information at a very high quality cannot be overestimated. Low-quality displays distract us by drawing our attention to pixelation, granularity, and poor representations.

The last decade has seen a remarkable improvement in display technology: most commercial laptop PCs now have a 13-inch color TFT (thin film transistor) LCD display at XGA resolution with a viewing angle of at least 140 degrees. However, these are mainly transmissive displays requiring a backlight that accounts for approximately one-third of the device's total power consumption. So, from a system viewpoint, these displays are far from ideal. Some LCDs have a resolution that exceeds 300 dpi, making them suitable for x-ray-quality pictures. Quality has also increased for very large displays, such as the 60-inch diagonal plasma display that Samsung showed at Comdex 2001. Such commercially available displays provide the means for shared display workspaces, which have been the subject of research for some time (for example, the Stanford I-Room[12]).

Related URLs

- Aibo: www.aibo.com
- Blackberry: www.blackberry.net
- Bluetooth: www.bluetooth.com
- DoCoMo: www.nttdocomo.co.jp (in Japanese)
- Electronic ink: www.eink.com
- IEEE 802.11 Wireless Local Area Networks: http://grouper.ieee.org/groups/802/11/index.html
- Infrared Data Association (IrDA): www.irda.org
- Intel StrongARM processors: http://developer.intel.com/design/pca/applicationsprocessors/index.htm
- Power MEMs research: http://web.mit.edu/aeroastro/www/labs/GTL/research/micro/micro.html
- Universal Plug and Play: www.upnp.org
- Versus Technology: www.versustech.com

At the smaller end, PDA displays have also improved. In the past year, PDAs such as the Compaq iPAQ have used transreflective color LCDs. These displays can take natural light entering their surface and reflect it back through the LCD stack, yielding considerable energy savings. Although all these displays look promising, they still have scope for improvement: the contrast ratio is lower than most printed magazines, and the resolution also needs to increase to the level of print media.

Ultimately, the need for improvement in display technology is bounded by the human eye's visual acuity, but we still have some way to go. A truly ubiquitous computing experience requires high-quality displays to let us see through the display process and effortlessly acquire the underlying information.

Adoption trends

For the year 2000, the annual sales of PCs were approximately 150 million units,[13] a remarkable statistic about the rate of adoption of computer technologies into all aspects of modern life. However, in that year 8 billion embedded processors made their way into the infrastructure of industry and electronic consumer devices. So, the fraction of PC sales is a mere 2 percent of the total processors sold. From this statistic, it is obvious that processors are beginning to be deployed ubiquitously.

Similarly, another market trend is the convergence of previously separate technologies. Devices such as cell phones, PDAs, and digital cameras are beginning to merge. Their combined capabilities reduce the number of devices a user must carry or own. Such convergent devices will likely succeed because numerous devices will no longer burden users.

In contrast, the problems that a unified device's size and complexity create have given rise to single-function *information appliances*. These appliances are easily adopted because of their simplicity and low cost. When personal mobility is not the main motivating factor, such divergent design approaches become attractive—for example, customizing a computer to be solely a spell checker. Such a device is physically better suited to its task, but then we have many diverse devices to keep track of and maintain. Even so, organized users can "accessorize" their devices for a particular task. There is an ongoing tension in ubiquitous-hardware design between convergent and divergent devices.

We can see that our first taste of ubiquitous computing is already in the PC's two strongholds—the office and home—and a unique third domain, the automobile. The office has benefited from an integration of applications and services driven by the need for coordinated enterprise solutions. For example, the Blackberry, a wireless device that integrates with corporate email, has created a considerable following among the corporations that have adopted it. Such examples show how business pressure is pushing the development of integrated systems, which are based on ubiquitous computing principles.

The home is also a natural driver for ubiquitous computing system design. We are at an early evolutionary stage where some homes have established a high-speed network connecting multiple PCs. Consumer products are emerging that exploit this infrastructure—the home computer system is slowly subsuming all the home communication, entertainment, information delivery, and control systems.

The automobile is a particularly outstanding area of success for ubiquitous computing. Modern cars have computer systems that integrate control of the engine, transmission, climate, navigation, entertainment, and communication systems. Success in this domain is largely because each car manufacturer has had complete control over all aspects of its subsystems, unlike the home and office domains. Also, power for computation is not a limitation because it is a small fraction of what is needed to accelerate the car.

Where we're headed

Ubiquitous computing focuses on getting computing "beyond the desktop." This immediately presents a set of research challenges associated with removing the customary stationary display-and-keyboard model, because for most people the PC is still their focus. Most current ubiquitous computing research projects fall into two categories: personal systems, which include mobile and wearable systems, and infrastructure systems, which are associated with a particular physical locale. For both categories, novel interaction modalities, such as speech or pen processing, become a necessary component because they don't require bulky displays or input devices.

Outside these two categories, the interaction of computers with the physical world, without direct human involvement, is rapidly becoming more important as the total amount of deployable computation increases. A system's ability to proactively monitor and react to the real world is instrumental for truly ubiquitous computing, where not only the hardware but also the results of actions must be removed from the foreground. Similarly, robotics is an emerging field that mobilizes a computer and enables it to effect change at arbitrary locations in the real world.

Personal systems

Personal systems give users access to computing independent of their physical location at the cost of them having to carry some equipment. Discrete portable devices such as PDAs and cell phones are currently the most useful personal systems. However, these systems tend to be limited by their computational ability, integration with other devices (for example, your cell phone communicating with your PDA), and interface capabilities (such as the display's size and quality). Today, computational ability and integration are not hindered by a fundamental limit—advances in processors and short-range wireless technology will eventually solve these problems—but interface capabilities are.

Some systems, typically termed *wearable computers*, rely on hardware such as heads-up displays and one-handed keyboards to provide the interface to the computer. This model is attractive because it provides a fully functional computing experience wherever the user might be. However, these interfaces can be overly intrusive, requiring a great deal of the user's attention; this mitigates their widespread acceptance. Currently, these devices are typically fairly bulky belt-worn devices, but they will shrink as technology progresses, thereby lending themselves to better industrial design and integration.

Personal servers[14] (see Figure 2a) can enhance ubiquitous access to data. They form the computation and storage center of a person's digital experience. Devices such as a cell phone or PDA-like appliance communicate directly with this central server, providing a common representation of a user's data. These devices, therefore, merely represent an interface into this central repository. With this model, the personal server could be located out of easy reach—for example, in a user's shoe, handbag, or belt clip—without causing inconvenience. Or, it could even be combined with an existing device (such as a cell phone).

Extending this model, a user's personal server could also support interaction through interfaces in the surrounding infrastructure. Users could access their personal data through a public kiosk or borrowed laptop display (see Figure 2b). This extended model is attractive because it allows a favorable user experience (interacting with personal data through a large display) without requiring users to carry the display. The personal server has recently become tractable owing to advances in short-range wireless technologies (for example, Bluetooth) and low-power processing (for example, StrongARM), which can now support an acceptable user experience.

Infrastructure systems

Unlike mobile systems, infrastructure systems instrument a particular locale. So, many of the difficulties shift from issues of size, weight, and performance to those of deployment, management, and processing. For example, imagine thousands of miniature temperature sensors deployed around a room—How did they get there? How is the data collected? How are faulty components identified and replaced?[15] Tasks that are tractable when the human/computer ratio is close to one suddenly become difficult when the number of computing devices increases.

Several hardware projects, such as the Berkeley motes,[16] have started to explore this space by creating a fairly small wireless sensor platform. This lets researchers actively explore the networking protocols necessary to organize large sets of nodes. Currently about the size of three stacked US quarter-dollar coins, these devices will keep shrinking until they reach the size of smart dust[17] and can no longer be seen or directly manipulated. The Berkeley Pico Radio project is pursing a single-chip system that incorporates both processing and radio frequency subsystems.[18]

Although the basic hardware of embedded devices can be relatively simple, considerable hardware challenges remain. Power is a primary concern. Even though

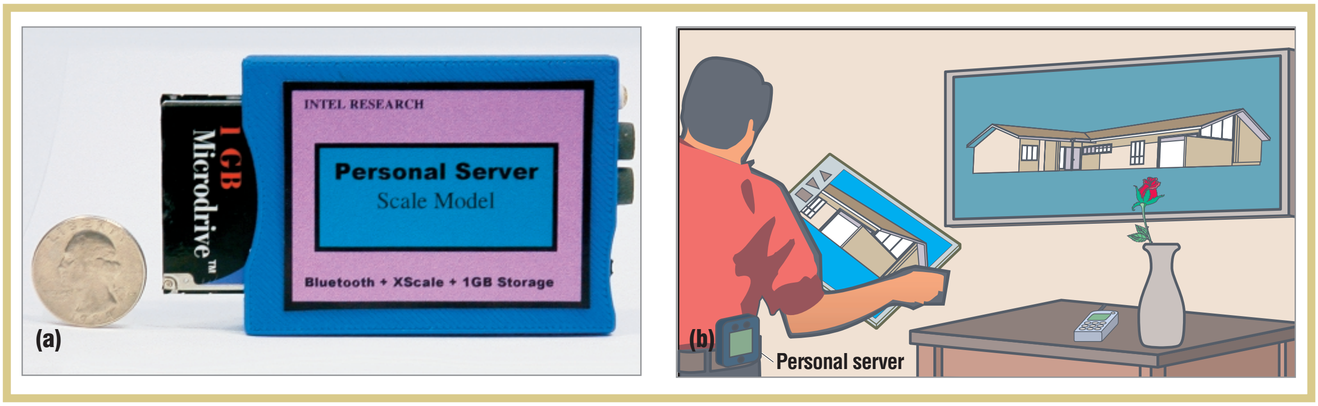


Figure 2. Personal servers: (a) Intel Research's prototype; (b) using resident display devices such as a wall-mounted display, or a community tablet computer, to view personal data stored on a user's personal server.

each node might have a battery lifetime of several months, the mean time to battery failure for the entire system can be short if it contains many devices. Furthermore, the environmental impact of these systems is not well understood—dropping a thousand sensing devices out of an airplane on a disaster zone is relatively easy, but how do you reclaim the devices?

The fundamental problem with shrinking hardware devices is that they quickly become too small and numerous for people to relate to them: they are literally out of sight, so they will quickly become out of mind. One direct byproduct of these multidevice systems is that the individual network nodes will not be named in any human-understandable way—there will simply be too many of them to keep track of. System security rapidly becomes a big concern: how do you know if a node should be listening in on a wireless conversation if you can't even keep track of which nodes you have in the system?

Proactive interaction

Both personal and infrastructure-based systems ultimately require some kind of user interface to let humans interact with them. Nondesktop interface modalities, such as pen, speech, vision, and touch, are attractive in ubiquitous computing systems because they require less of a user's attention than a traditional desktop interface. However, the mainstream use of these techniques depends largely on their hardware requirements and interface capabilities.

For example, pen computing has largely been successful on PDA-class devices with a well-defined and accessible display. But it has no place in systems with either very small displays or no display at all. Speech and vision interfaces have done well in infrastructure-based systems that have both significant computation resources and a static environment.[19] But they have trouble in mobile systems that are computationally impoverished or need to operate in dynamic environments that might be, for example, very loud or dark. Complex touch-based interfaces[20] that deal with a significant amount of data input or output are problematic. They tend to be task specific, with interface hardware crafted for a specific application, so they have not gained wide acceptance.

The most exciting advances in ubiquitous interfaces will likely be new display technologies that enable rich visual output without a bulky flat-screen display. E-Ink (electronic ink) Corp., for example, uses a system of microcapsules to create flexible display surfaces, which would be considerably more convenient than a traditional rigid LCD panel. Such technology, in conjunction with an abstracted computation model such as the personal server, brings us one step closer to a world where we can access personally relevant information quickly and conveniently, without relying on bulky, fragile display systems. Interface hardware technology poses different difficulties for mobile systems than for desktop computers. For example, speech interfaces for mobile devices are inherently problematic because they often operate in noisy environments, requiring noise cancellation or directional-microphone techniques.

Directly integrating computing with the real world, without human intervention, will greatly increase computers' impact on our lives by removing people as a major limiting factor from the processing stream. A few automatic systems have already significantly affected our lives: thermostats, airplane autopilots, automated factories, and antilock brakes, for example. Such systems still require human intervention: thermostat maintenance, airplane landings, factory construction, and deciding when to apply the brakes. The challenge is to make these systems proactive, where they can anticipate and react to physical world conditions (for example, deciding to apply the brakes), instead of just reacting to them (for example, deploying an air bag when you crash).[21] The salient distinction between these two models is that one is human-centric, which requires close involvement to effect correct operation, while the other is human-supervised, requiring minimum involvement but still achieving an intelligent, useful result.

A proactive system must closely and reliably integrate sensors and actuators with the physical world. This task is closely related to building the infrastructure-based systems we described earlier. However, proactive systems will require greater sophistication in the components deployed in the environment, both to enable the capability to affect the physical world and to quickly, robustly, and accurately process real-world data. For example, for a building-scale temperature-monitoring application, slowly reporting distributed temperatures to a central server would be sufficient. However, an earthquake response system would need to actively dampen vibrations at many nodes throughout a building.

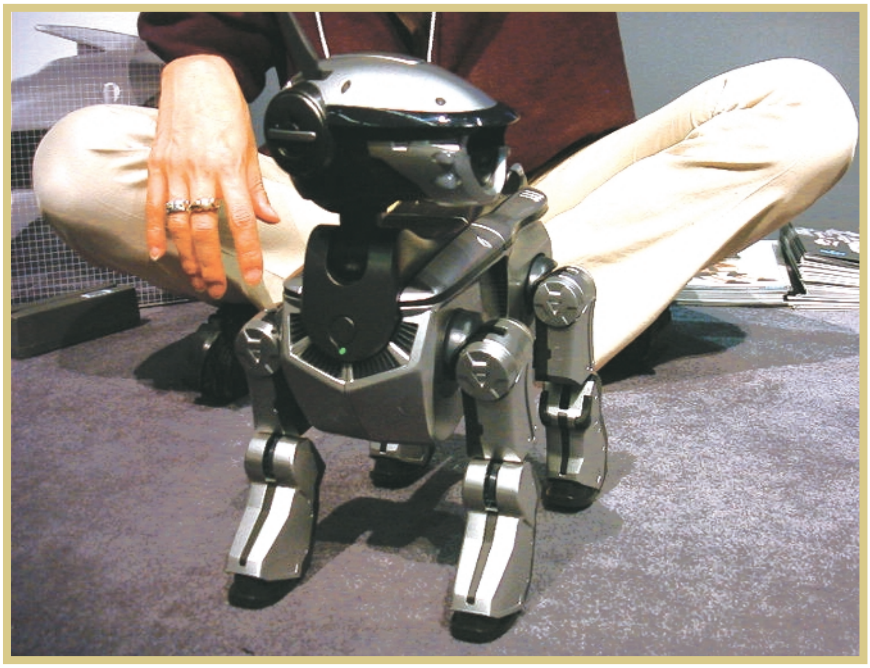


Figure 3. Sony's Aibo robot dog.

From a personal perspective, proactive computing pushes us toward systems that can monitor and affect our bodies directly. For example, automatically handling diabetes requires the ability to both monitor blood glucose levels and administer insulin—tasks requiring specialized hardware. A system such as this would still require significant advances in software reliability to be feasible.

Robotics

Now more popular than ever, robotics presents an interesting confluence between mobile and proactive systems: allowing a computer system to affect the real world without a priori instrumentation of the environment. Most robotic systems either perform a very specific task, such as factory automation, or are remotely controlled by a person. In general, autonomous robots have difficulties dealing with dynamic physical situations—primarily with determining where they are, as well as with identifying objects in the environment. Similar to handheld devices, the hardware for constructing robots is shrinking rapidly, making them cheaper and more capable. Recently, the first real consumer robots—namely, Sony's Aibo (see Figure 3)—have reached the market. Abstractly, robots are a novel type of disappearing hardware: they allow computation to directly affect the real world without heavily instrumenting the environment.

Software issues aside, significant hardware challenges exist, particularly for the sensing technologies that will enable proactive robotics. Vision is the canonical technique for a robot to determine where it is, what objects are around it, and where they are. The generalized vision problem required to deal with dynamically changing physical locations is quite difficult. One solution is to instrument a particular environment with sensors, beacons, or both to aid a robot in a particular locale—the equivalent of GPS for buildings. A high-accuracy indoor-positioning system would better allow an autonomous robot to function within a building's confines. Additionally, tagging interesting objects in the environment would help a robot identify and locate them. By exploiting infrastructure-based computing, as we discussed earlier, robotics can make significant strides toward being truly proactive and autonomous without solving the generalized location problem mentioned in the previous paragraph.

Hard problems & physical limitations

Three hard problems have been the core challenge of ubiquitous computing hardware and will likely continue to be far into the future: size and weight, energy, and the user interface. These problems transcend individual points on the technology curve, partly because they are somewhat contradictory—as we'll see, a solution in one space greatly confounds that in another. Ongoing solutions to these problems, therefore, will have to come from "outside the box" and remove the fundamental roadblocks.

Size and weight

Size and weight significantly impede ubiquitous computing because they continually remind the user of the hardware's presence. This limitation manifests itself both in mobile systems when the user must carry the device and during the setup and configuration of infrastructure-based systems involving many pieces of equipment. Appropriately, the two main contributors to a device's size and weight stem from the other two fundamental problems: batteries and the user interface.

The Itsy pocket computer (see Figure 4)[22] exemplifies this situation. It is just 70 percent larger than its 2.2 watt-hour battery and 320- × 200-pixel display. In addition, these two components represent 60 percent of the total weight without the case but 43 percent with the case.

Overall, device sizes have been shrinking at an incredible rate owing to system-level integration. But devices are reaching the point where they can't get much smaller and still be usable or provide additional benefit. Moreover, such reductions might increase, rather than decrease, the cognitive load. For example, many new cars include a key fob for locking the doors and arming the alarm. If the fob were the size of a brick, few people would use it because it would be too large. Alternatively, if it were the size of a penny, few people would use it owing to difficulty manipulating the

Figure 4. The Itsy pocket computer exemplifies the trend that the size and weight of energy sources and I/O devices is dominating the size and weight of mobile computers.

buttons. In this case, the most suitable size and weight are inextricably tied to the device's intended function.

Energy

The challenge for any mobile system is to reduce user involvement in managing its power consumption. Energy is a necessary resource for virtually all computing systems, but any reference to it detracts from a positive user experience. The degree to which an energy source distracts the user depends on the intended application, the hardware implementing the application, and the energy source's characteristics.

Solutions to this problem fall into two approaches. The first is to reduce power consumption. Part of this approach consists of designing energy-aware software that can identify the hardware states that provide a given service level and select those that are most energy efficient. For instance, in systems with a microprocessor whose energy consumption is greater at high speeds, the software can select the lowest speed possible that still achieves the required task's performance.[23] The control software can also modify the quality of service it seeks to deliver.[24] For instance, to save energy, the software could reduce the frame rate, or size, of an MPEG movie, incrementally resulting in a corresponding loss of fidelity. Or, the software could forward a voice utterance to a remote system for recognition rather than expending local energy on the task.[24] Software systems are just beginning to address these issues concerning energy awareness.

The second approach is to find alternative and improved energy sources. Although battery energy densities are projected to increase approximately 10 percent annually for the next three years,[25] the energy storage costs of batteries will likely remain significant. This will lead us to explore other technologies for storing, and even generating, energy. For example, MIT researchers have exploited the human body as an energy source by constructing sneakers that use flexible piezoelectric structures to generate energy (see Figure 5).[26,27] Similarly, solar radiation, thermal gradients, mechanical vibration, and even gravitational fields all represent potential power sources for a mobile device.[15]

Additionally, storage technologies under development promise much greater energy densities than those of conventional batteries. In the near term, one of the more promising technologies is fuel cells, particularly direct methanol fuel cells.[28] Pure methanol fuel offers an energy density roughly 40 times that of a Lithium-ion polymer battery, but 70 to 90 percent of this chemical energy is lost in conversion to electricity. In the longer term, technologies such as MIT's MEMS (microelectromechanical systems)-based microturbine and associated micro electric generator might provide highly compact energy sources with significantly longer lifetimes.

An alternative to acquiring energy is to transmit energy to a mobile device, reducing its need for an autonomous power source. This technique is difficult to do safely over a long range, but it is applicable to the field of passive electronic tagging. For example, *Radio Frequency Identification*[29] tags are inductively powered by the tag reader, typically up to a maximum of one meter, employing load modulation to transmit their data back to the interrogator. Such passive tags have unlimited lifetimes, are smaller, and cost less; however, unlike battery-powered (active) tags, they can communicate only over a short range and cannot autonomously signal their presence. The ability to transmit energy, when combined with robotics, gives us the capability for mobile computation that can recharge itself.

User interfaces

Rendering user interfaces invisible is fundamentally difficult owing to the tradeoff between size and weight and usability. Reducing the size and weight will make the device less visible but might decrease its usability. The degree to which these two components matter depends on the specific properties of the interface and of the application for which it is being used.

For example, to open or lock a door, a user might use a remote control containing one button that unlocks and opens the door when pressed once and locks it when pressed twice in quick succession. Because the single button serves multiple purposes, users will likely find it more demanding than the original interfaces (the door knob and door lock). Alternatively, the remote control could have two buttons, one for opening and one for locking and unlocking the door. At the expense of increased size and weight, this solution substitutes a spatial differentiation (two single-function buttons) for a temporal one (a multiple-function button). These two solutions represent the chief user-interface tradeoff that makes good user interface design for small ubiquitous devices fundamentally hard.

Today's most popular user interfaces include buttons, keyboards, mice, pointers, LCD panels, touch screens, microphones, and speakers. These elements are designed for high-rate information flow, each suited to a specific application class. For example, a keyboard and display seem ideal for writing a book, but a microphone and speaker would probably be better for communicating with another person. For other applications, such as alerting a user that he or she has received mail, these user interfaces are unnecessarily complex. Other, more unobtrusive, mechanisms such as the ambientROOM,[30] explore how to communicate low-latency, low-importance information via interfaces that require little direct attention. For example, variances in a projected image's intricacy could indicate if there is unread mail, thereby making information available to the user unobtrusively through a directed glance.

Although these interfaces are distinctly separate from our bodies, the natural direction for disappearing interfaces is for the two to blend together. Several researchers are exploring the possibility of interpreting information from our neurons to let us control computers and other machines by just thinking about doing so.[31] Early work involving a monkey with neural implants has demonstrated the ability to gather sufficient information to let a robot mimic the monkey's arm movement.[31] Similarly, neural implants can feed information into the brain, removing the need for humans to gather information through their senses. This approach's potential is suggested by recent research employing cochlear implants to help those with hearing loss to communicate better and become more aware of their surroundings.[32] For example, such neural implants could bring information (such as "you've got mail") to the user's attention by fooling the brain into thinking a subtle but noticeable image has been projected onto the retina.

These interfaces form the personal-interaction version of proactive computing, where the computer and the real world are tightly integrated. Nonetheless, direct neural interfaces embody tremendous risks, not the least of which is loss of human autonomy. Clearly, this area requires much more research, but if we can overcome the challenges, such interfaces would go a long way toward reaching Weiser's vision.

Future challenges

We will soon be able to include computer hardware into virtually every manufactured product, and provide a wireless infrastructure to let these devices communicate directly or indirectly. But what will they communicate about and in what protocol? How will a user or user's application know what devices to use for what purpose? How will a user know when invisible devices are present, functioning, and not compromising privacy? What actions can a user take if the situation is unacceptable? These questions suggest that hardware, software, user interaction, and applications all have unresolved issues that we must address before ubiquitous computing will truly reach Weiser's goal of improving, rather than further complicating, our lives.

A key element of ubiquitous computing applications is knowing the precise spatial–temporal relationships between people and objects.

Unresolved hardware issues

We have already discussed some of the challenges for our hardware platforms. Specifically, we will need to continue to manage our devices' power requirements. Making devices ubiquitous can't be coupled with the need to change batteries, or even recharge them, when thousands of devices are involved. The solutions presented earlier only partially solve the problems; some are highly experimental and might not be practical for unforeseen reasons.

Microprocessors can also continue evolving. The need will persist for minimal devices that use the baseline minimum power for the job or occupy the smallest possible volume. Processors will continue to shrink while increasing in capability and capacity. However, new applications will demand ever-greater processing capabilities, not the least of which will be those of communication and security. For example, implementing public key encryption with a typical microcontroller instruction set leads to large code size, high power requirements, and slow performance.[33] The challenge will be to include appropriate primitives that make these operations more efficient in all these dimensions while permitting the evolution of security algorithms.

As we crowd more and more frequencies with wireless communications, we will need to make our radio systems adaptive. Devices will need to adjust their bandwidth requirements on the basis of what other devices are in the radio neighborhood. Such a necessity highlights the critical problem of system evolution. How will the development and deployment of new devices affect the devices we already have deployed? Flexibility will be needed, not only in terms of software updates to adjust how a device is used but also in terms of wireless communication. Software radios are promising in this second dimension; they let a device change how it uses the spectrum to be compatible with its neighbors.[34]

Disappearing software

To make hardware disappear, we also need to make software disappear. Today's software is too monolithic and stovepiped, and is written with many assumptions about hardware and software resources. The ubiquitous computing environment that we must create will be much more dynamic. Devices, objects, and people will be constantly moving around, creating an ever-changing set of resources—different user input/output interfaces, displays, and windowing systems—that will be available to our applications. Also, some devices might lose their ability to communicate with others owing to interference or environmental conditions. How will we construct applications to operate in such environments?

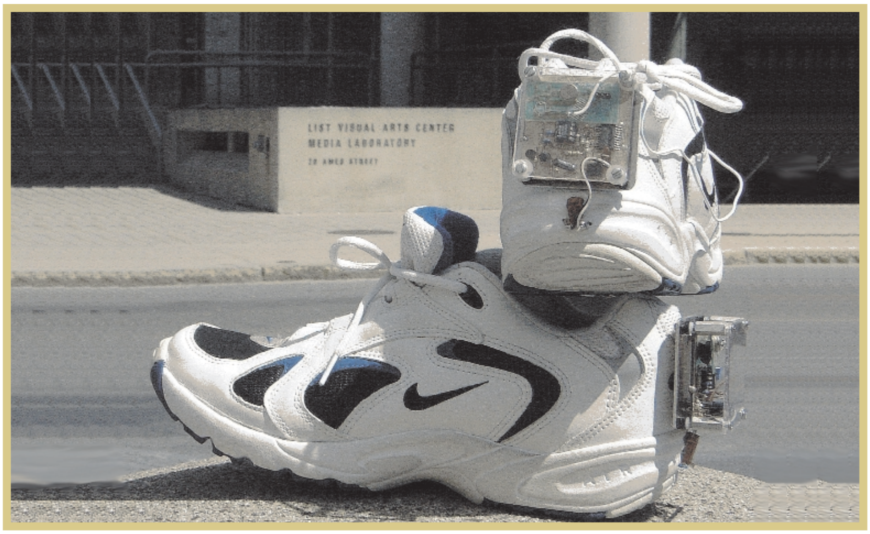


Figure 5. Energy-scavenging shoes using piezoelectric material, developed at MIT's Media Laboratory. With these shoes, normal walking motion can generate sufficient energy to broadcast an ID every three to five steps.[27] Photo courtesy of the MIT Media Laboratory.

To do this, we must even further decouple our applications into small pieces of code, spread across ubiquitous hardware, that can come together as needed and expect to have connections constantly enabled and disabled. We must create data interchange formats where the data not only is self-describing but also can find its way from the device that created it to its destination as autonomously and securely as possible.[35] We must develop mechanisms for devices to advertise their capabilities so that applications, whether in the infrastructure or on portable devices, can become aware of and select from the panorama of available resources.

Discovery services[36,37] (for example, Universal Plug and Play) to some extent already enable such *cyber foraging*.[38] These first-generation systems require highly capable devices that can download code and enjoy stable and relatively high-bandwidth connections. This approach will not scale to the myriad sensing and processing devices that will surround us. They will employ minimal computational elements and exist in small communication cells with highly variable communication and processing properties. This scenario requires that multiple devices replace a single device. Adaptation at this level is another challenge we must tackle to achieve long-lived systems that can evolve gracefully.

Interaction design

Although new technologies are creating new ways of interacting with our computations, we could end up trading the problems of one user interface for those of another. For example, wireless technologies permit a device we are carrying to interact with a public display. But what happens when multiple potential users are surrounding a display? Not only must the software handle multiple users, but also the hardware must be able to accurately identify and differentiate between the multiple users. Addressing these issues using what we've learned from the desktop metaphor seems less promising because that interface has not been developed or optimized for casual use by multiple users.

With computing capability in every object, users will want to take advantage of the devices they encounter throughout their day without worrying about ownership or security at every step. Consider how we use paper and pencil: we can easily jot down notes and later identify the author based solely on the handwriting. The note-taking process directly incorporates this process; there is no explicit authentication mechanism. Similarly, physical control guarantees privacy: we put the paper in our pocket and can hide it from others' eyes. What metaphors will we have for the electronic paper that can communicate with other devices? Without some kind of physical icon, how do we seamlessly control sharing content with other people?

A key element of ubiquitous computing applications is knowing the precise spatial–temporal relationships between people and objects. Such knowledge succinctly helps specify intent, an integral component of user interfaces. But the resolution of location systems needs to improve dramatically.[39] Current commercial indoor location systems are either too coarse, operating primarily at the room level[7] (for example, the Versus Information System), or require prohibitively expensive infrastructure,[40,41] or are based on proximity (for example, electronic tags). We desperately need tags that can be located within a few millimeters, are cheap to create (for example, by printing), and are completely passive. Ideally, we would also want the ability to detect that two tagged objects are physically touching rather than just in close proximity.[42,43] Coming up with the technologies that provide these capabilities, even if initially imperfectly, is another key challenge. For more on this topic, see "Connecting the Physical World with Pervasive Networks," in this issue.

Applications

Currently, most applications are based on ownership of relatively large, multipurpose hardware devices with only the most limited interaction with the physical world in which people live. We need to discover and enable those most compelling applications that will let us deploy the first truly ubiquitous systems. Yet, as we have already discussed, the most difficult part of this will be to give these systems an evolutionary path, in contrast to today's approach of completely reengineering all the component devices. Software engineering will take center stage in this effort. Clearly, we need a new set of abstractions to make writing such applications possible—abstractions that support software deployment in separate modules on thousands of devices, cyber foraging, and hardware sharing. This is a different world of software development than we are accustomed to.

Many hardware components necessary to build ubiquitous computing systems are now available. Key improvements since Weiser's original vision of ubiquitous computing include wireless networks; high-performance and low-power processors; high-quality displays; and high-capacity, low-power storage devices. The progress toward building software systems to orchestrate these components has been less dramatic, with many unresolved issues relating to user interfaces, security, privacy, and managing complexity. Consequently, the hardware components for many applications are reaching the point where a user is less likely to be distracted by the medium than by the interaction with the controlling software. And as a result, at this level, the hardware is beginning to become invisible.

Of course, hardware could become too invisible. At some point, you might need to know what is real and what is simulated, and have some handle on an interface's location and components. However, technological advances clearly will continue to bring us new hardware components that, if we wish, will function more invisibly than before, until every aspect of an interface crosses a threshold that no longer hinders our senses. We will begin to see through it, just as we see through the ink of printed text and focus on the information it contains. At that point, by definition, the hardware will have disappeared. When that day comes, computer hardware will likely be mediating in every aspect of our daily activities, and Weiser's vision will be almost complete.

ACKNOWLEDGMENTS

The thoughts and views expressed in this article were not formed in isolation. We thank the many colleagues who have worked alongside us on ubiquitous computing projects and, in doing so, have shaped our views in this exciting research area. Particularly, we thank Sunny Consolvo, Gunner Danneels, and the *IEEE Pervasive Computing* editorial board and supporting reviewers for their comments; Jim Kardach, Graham Kirby, Muthu Kumar, Murali Sundar, and Paul Wright for the presentation of the personal server prototype; and Bob Mayo and Bill Hamburgen for their contributions to our discussion on the hard problems associated with ubiquitous computing. And a special thank you to Mark Weiser, in memory, for his visionary ideas in the early 1990s.

REFERENCES

1. K. Fishkin et al., "Embodied User Interfaces for Really Direct Manipulation," *Comm. ACM*, vol. 43, no. 9, Sept. 2000, pp. 75–80.
2. M. Weiser and J. Seely-Brown, "The Coming Age of Calm Technology," *Beyond Calculation: The Next Fifty Years of Computing*, P.J. Denning and R.M. Metcalfe, eds., Copernicus, Heidelberg, Germany, 1998.
3. D.A. Norman, *The Invisible Computer*, MIT Press, Cambridge, Mass., 1998.
4. R. Want et al., "An Overview of the ParcTab Ubiquitous Computing Experiment," *IEEE Personal Comm.*, vol. 2, no. 6, Dec. 1995, pp. 28–43.
5. C. Kantarjiev et al., "Experiences with X in a Wireless Environment," *Proc. Usenix Symp. Mobile & Location-Independent Computing*, Usenix Assoc., Berkeley, Calif., 1993, pp. 117–128.
6. S. Elrod et al., "Liveboard: A Large Interactive Display Supporting Group Meetings, Presentations, and Collaborations," *Proc. 1992 Conf. Human Factors in Computing Systems* (CHI 92), ACM Press, New York, 1992, pp. 599–607.
7. R. Want et al., "The Active Badge Location System," *ACM Trans. Information Systems*, vol. 10, no. 1, Jan. 1992, pp. 91–102.
8. T. Truman et al., "The InfoPad Multimedia Terminal: A Portable Device for Wireless Information Access," *IEEE Trans. Computers*, vol. 47, no. 10, Oct. 1998, pp. 1073–1087.
9. B. Schilit, N. Adams, and R. Want, "Context-Aware Computing Applications," *Proc. Workshop Mobile Computer Systems and Applications*, IEEE CS Press, Los Alamitos, Calif., 1994, pp. 85–90.
10. G. Moore, "Cramming More Components onto Integrated Circuits," *Electronics*, vol. 38, no. 8, 19 Apr. 1965, pp. 114–117.
11. J.W. Toigo, "Avoiding a Data Crunch," *Scientific American*, vol. 282, no. 5, May 2000, pp. 58–74.
12. A. Fox et al., "Integrating Information Appliances into an Interactive Workspace," *IEEE Computer Graphics & Applications*, vol. 20, no. 3, May/June 2000, pp. 54–65.
13. *Embedded Microcomponent Forecast and Trends through 2004, MPUs, MCUs, DSPs and Cores*, Gartner Group, Stamford, Conn., 22 Jan. 2001.
14. R. Want and B. Schilit, "Guest Editors' Introduction: Expanding the Horizons of Location-Aware Computing," *Computer*, vol. 34, no. 8, Aug. 2001, pp. 31–34.
15. A. Chandrakasan et al., "Design Considerations for Distributed Microsensor Systems," *Proc. 1999 IEEE Custom Integrated Circuits Conf.*, IEEE Press, Piscataway, N.J., 1999, pp. 279–286.
16. J. Hill et al., *System Architecture Directions for Network Sensors, Proc. 9th Int'l Conf. Architectural Support for Programming Languages and Operating Systems* (ASPLOS 2000), ACM Press, New York, 2000, pp. 93–104.
17. J.M. Kahn, R.H. Katz, and K.S.J. Pister, "Next Century Challenges: Mobile Networking for 'Smart Dust,'" *Proc. ACM/IEEE Int'l Conf. Mobile Computing and Networking* (MobiCom 99), ACM Press, New York, 1999, pp. 271–278.
18. J.M. Rabaey, "Wireless beyond the Third Generation—Facing the Energy Challenge," *Proc. 2001 Int'l Symp. Low Power Electronics and Design* (ISLPED 01), ACM Press, New York, 2001, pp. 1–3.
19. J. Crowley, J. Coutaz, and F. Berard, "Perceptual User Interfaces: Things That See," *Comm. ACM*, vol. 43, no. 3, Mar. 2000, pp. 54–64.
20. H. Ishii and B. Ullmer, "Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms," *Proc. 1997 Conf. Human Factors in Computing Systems* (CHI 97), ACM Press, New York, 1997, pp. 234–241.
21. D. Tennenhouse, "Proactive Computing," *Comm. ACM*, vol. 43, no. 5, May 2000, p. 43.
22. W.R. Hamburgen et al., "Itsy: Stretching the Bounds of Mobile Computing," *Computer*, vol. 34, no. 4, Apr. 2001, pp. 28–36.
23. M. Weiser et al., "Scheduling for Reduced CPU Energy," *Proc. 1st Symp. Operating Systems Design and Implementation*, Usenix, Berkeley, Calif., 1994, pp. 13–23.
24. J. Flinn, *Extending Mobile Computer Battery Life through Energy-Aware Adaptation*, PhD dissertation, tech. report CMU-CS-01-171, Computer Science Dept., Carnegie Mellon Univ., Pittsburgh, 2001.
25. Y. Kim, "Capacity Enhancement Technology of Samsung Soft Case Li-ion Batteries," *Proc. Power 2001: 9th Ann. Int'l Conf. Power Requirements for Mobile Comput-*

ing, Wireless Electronic Devices & Business/Consumer Applications, 2001.

26. T. Starner, “Human-Powered Wearable Computing,” *IBM Systems J.*, vol. 35, nos. 3–4, 1996, pp. 618–629.

27. N. Shenck and J. Paradiso, “Energy Scavenging with Shoe-Mounted Piezoelectrics,” *IEEE Micro*, vol. 21, no. 3, May/June 2001, pp. 30–42.

28. S. Thomas and M. Zalbowitz, *Fuel Cells—Green Power*, Los Alamos Nat’l Lab, Los Alamos, N.M., 1999; www.lanl.gov/energy/est/transportation/trans/pdfs/fuelcells/fc.pdf.

29. K. Finkenzeller, *The Radio Frequency Identification (RFID) Handbook*, John Wiley & Sons, New York, 1999.

30. H. Ishii et. al., “AmbientROOM: Integrating Ambient Media with Architectural Space,” *Proc. 1998 Conf. Human Factors in Computing Systems* (CHI 98), ACM Press, New York, 1998, pp. 173–174.

31. A. Regalado, “Miguel Nicolelis: Brain-Machine Interfaces,” *Technology Rev.*, Jan./Feb. 2001, pp. 98–100.

32. M. Nicolelis, “Actions from Thoughts,” *Nature*, vol. 409, no. 6818, 18 Jan. 2001, pp. 403–407.

33. A. Perrig, “SPINS: Security Protocols for Sensor Networks,” *Proc. 7th Ann. ACM Int’l Conf. Mobile Computing and Networking* (MobiCom 2001), ACM Press, New York, 2001, pp. 189–199.

34. V. Bose, D. Wetherall, and J. Guttag, “Next Century Challenges: RadioActive Networks,” *Proc. ACM Mobile Computing and Networking* (Mobicom), ACM Press, New York, 1999, pp. 242–248.

35. M. Esler et al., “Next Century Challenges: Data-Centric Networking for Invisible Computing—The Portolano Project at the University of Washington,” *Proc. MobiCom 1999*, ACM Press, New York, 1999, pp. 256–262.

36. W. Adjie-Winoto et al., “The Design and Implementation of an Intentional Naming System,” *Operating Systems Rev.*, vol. 34, no. 5, Dec. 1999, pp. 186–201.

37. J. Waldo, *Jini Architecture Overview*, tech. report, Sun Microsystems, Palo Alto, Calif., Jan. 1999.

38. M. Satyanarayanan, “Pervasive Computing: Vision and Challenges,” *IEEE Personal Comm.*, vol. 8, no. 4, Aug. 2001, pp. 10–17.

39. J. Hightower and G. Borriello, “Location Systems for Ubiquitous Computing,” *Computer*, vol. 34, no. 8, Aug. 2001, pp. 57–66.

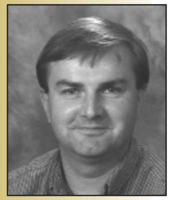
40. M. Addlesee et al., “Implementing a Sentient Computing System,” *Computer*, vol. 34, no. 8, Aug. 2001, pp. 50–56.

41. A. Ward, A. Jones, and A. Hopper, “A New Location Technique for the Active Office,” *IEEE Personal Comm.*, vol. 4, no. 5, Oct. 1997, pp. 42–47.

42. K. Partridge et al., “Fast Intrabody Signaling: Empirical Measurements of Intrabody Communication Performance under Varied Physical Configurations,” *Proc. 14th Ann. ACM Symp. User Interface Software and Technology* (UIST 2001), ACM Press, New York, 2001, pp. 183–190.

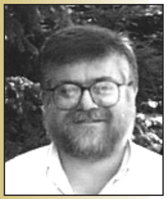
43. B. Vigoda and N. Gershenfeld, “TouchTags: Using Touch to Retrieve Information Stored in a Physical Object,” *Extended Abstracts of Computer-Human Interaction 1999*, ACM Press, New York, 1999, pp. 264–265; www.media.mit.edu/physics/publications/papers/99.01.touchtag.rtf.

the AUTHORS



Roy Want is a principal engineer at Intel Research. His interests include ubiquitous computing wireless protocols, hardware design, embedded systems, distributed systems, and automatic identification and microelectromechanical systems. While at Olivetti Research, he developed the Active Badge, a system for automatically locating people in a building. As part of Xerox PARC’s Ubiquitous Computing program, he led the ParcTab project, one of the first context-aware computer systems. At PARC, Want also managed the Embedded Systems group. He received his BA and PhD in computer science from Churchill College, Cambridge University. Contact him at Intel Corp., 2200 Mission College Blvd., Santa Clara, CA 95052; roy.want@intel.com; www.ubicomp.com/want.

Trevor Pering is a research scientist at Intel Research. His research interests include many aspects of mobile and ubiquitous computing, including usage models, power management, novel form factors, and software infrastructure. He received his PhD in electrical engineering from the University of California, Berkeley, with a focus on operating system power management. He is a member of the ACM. Contact him at trevor.pering@intel.com.



Gaetano Borriello is a faculty member of the University of Washington’s Department of Computer Science and Engineering. He is on a two-year leave to establish a new Intel research center adjacent to the University of Washington campus. His research interests are in the design, development, and deployment of computing systems with particular emphasis on mobile and ubiquitous devices and their application. His most recent research accomplishments include the development of the Chinook design system for heterogeneous distributed embedded processors. He received his BS in electrical engineering from the Polytechnic Institute of New York, his MS in electrical engineering from Stanford University, and his PhD in computer science from the University of California, Berkeley. He received an NSF Presidential Young Investigator Award in 1998 and a University of Washington Distinguished Teaching Award in 1995. Contact him at the Dept. of Computer Science & Eng., Univ. of Washington, Box 352350, Seattle, WA 98195-2350; gaetano@cs.washington.edu.

Keith I. Farkas is a senior member of the research staff of Compaq Computer’s Western Research Lab. His research interests include microprocessor design and software and hardware techniques for managing and optimizing computer system energy consumption. He received his PhD from the University of Toronto. He is a member of the IEEE and ACM. Contact him at keith.farkas@compaq.com.