4 Embedded Gear

Universal	Situated
Anytime-anyplace	Responsive place
Mostly portable	Mostly embedded
Ad hoc aggregation	Accumulated aggregation
Context is location	Context is activity
Instead of architecture	Inside of architecture
Fast and far	Slow and closer
Uniform	Adapted

4.1 Universal versus situated computing

When everyday objects boot up and link, more of us need to understand technology well enough to take positions about its design. What are the essential components, and what are the contextual design implications of the components? How do the expectations we have examined influence the design practices we want to adapt? As a point of departure, and with due restraint on the future tense, it is worth looking at technology in itself.

To begin, consider how not all ubiquitous computing is portable computing. As a contrast to the universal mobility that has been the focus of so much attention, note the components of digital systems that are embedded in physical sites (figure 4.1). "Embedded" means enclosed; these chips and software are not considered computers. They are unseen parts of everyday things. "The most profound technologies are those that disappear," was Mark Weiser's often-repeated dictum. "They weave themselves into the fabric of everyday life until they are indistinguishable from it."[1]

Bashing the Desktop

Some general background on the state of the computer industry is necessary first, particularly regarding the all-too-familiar desktop computer. By the beginning of the new century, technologists commonly asserted that a computer interface need not involve a keyboard, mouse, and screen.[2] Ambient, haptic, and environmentally embedded interface elements have become viable, and these require more concern for physical context. But because the desktop graphical user interface was just what made computing accessible to nonspecialists, it remained the only form of human-computer interaction that most peo- ple had ever known. By now that model is more than twenty years old. In the time frame of the information industries, that is astonishingly long.[3]

"What's wrong with the PC?" usability advocate Don Norman asked. "Everything. Start with the name. The personal computer is not personal nor is it used to do much computing."[4] Technology experts generally agree that the personal computer was a good idea for its time—twenty years ago. But by now that box is used for so many pur- poses that it has become unwieldy for any of them. Even its most essential applications—text, graphics, databases, and spreadsheets— have been weighed down under hundreds of commands. A graphical user interface made sense when you could put everything of relevance on the screen, but now there is just too much to see. A single hard disk made sense when its owner could know more or less what was on it, but now it typically contains thousands of files, most of them put there by someone else.

Software piles up. Programmers' impetus to add ever more fea- tures has led to bloat. This prolixity is a consequence of logic, which is naturally cumbersome. Each time another option is considered, the number of conditions that must be coded and configured increases. Faster chips run all this code acceptably, but people and organizations seldom benefit.

This lamentable state of the desktop computer results from poor design. As software critics such as Alan Cooper have explained, design for usability or experience too often comes only after this engineering, by which time it is too late.[5] As long as people accept design to be some programmer's notion of what features to add this year, there will always be more features, whether or not they are usable or useful.

And now, as computers move out into the physical world, better design becomes essential. Pervasive computing has been hailed as an escape from the desktop and a chance to start over. On the other hand, unless design can intervene, it is also a chance for computer technolo- gy to become even worse, and far less escapable.

The New Form of Locality

For any new approach to design to break out of this feature-accumu- lation cycle, information technology must change fundamentally, that is, at a level much more basic than a better desktop interface. In essence we face limits to how much we care to do or will consider doing with any one device in one place. More subtly, we also face limits to how much a device can do without better information about its context.

In response, the computer industry now researches aggregations of smaller, more specialized, more localized systems. As Sun Microsystems chief scientist Bill Joy has argued, we are actually facing the evolution of operating systems. The next stage of evolution takes the load off a technology now two paradigms old. Thing-centered computing is coming to be for the 2000s what network-centered com- puting was to the 1990s and personal computing was to the 1980s. A personal computer was (and still mostly is) based on its hard disk. The disk operating system (DOS) on which this was based assumed that everything one needed was stored locally. A network of computers, on the other hand, was (and presumably will remain) based on packet- switched communications. The transmission protocol (TCP/IP) on which this was based assumed that everything one needed could be made universally accessible on the Internet. This arrangement did not do away with the local hard disk, but it moved information on and off it in a way that was much more significant. Whereas the isolated desk- top system encouraged only more automation of tasks, network com- puting allowed full-scale organizational change. So many possibilities had to be accounted for, however, that the old disk operating systems grew into baggy monsters.

This evolution intrinsically embraces context. DOS assumed every- thing was just local (but not at all networked). TCP/IP assumed connec- tivity was universal. What sort of standard will emerge on the assump- tion that what you need, and with whom you wish to be connected at the moment, is based on where you are? Bill Joy called this a question of etiquette. If our growing constellations of devices and gadgets are to become any less obnoxious than the desktop computers they are intend- ed to replace, they will have to acquire some situational protocols.

This view often finds analogies in electrification. One usual cliché describes the refrigerator: You probably don't think of this machine as an application of an electric motor, and you don't need to know about electric motors to use it; you just treat it as a cold, dark place to store your food. Similarly "information appliances" let you carry out particular activities without having to be aware of any computers that may be involved. As Don Norman urged, "The primary motivation behind the information appliance is clear: simplicity. Design the tool to fit the task so well that it becomes part of the task, feeling like a natural extension of the work, a natural extension of the person."[6]

Norman's calls for "invisible" computers reflect the need for less attention-consuming technology. This echoes Mark Weiser's vision as well. "Invisibility has become the main issue now," he remarked in the early 1990s. "We have been very good at putting computers into the environment, but we have been very bad at getting them out of the way."[7]

The goal of natural interaction drives this movement toward pervasive computing and embedded systems (figure 4.2). There has been something unnatural about how desktop computers ignore many possible dimensions of skill and yet monopolize our attention. In a natural interaction, context would include information that does not require our attention except when necessary. Ambient interfaces would let us monitor potentially relevant information. Haptic and tangible interfaces would use latent intuitive physics.

Yet there is a lot more to embedded systems than there is to household electric appliances. This is basically a consequence of how they tend to link up. Ad hoc physical aggregations of digital devices become systems in themselves. Interoperability becomes critical here. "A distinguishing feature of information appliances is the ability to share information among themselves," Norman noted. "Smart devices are all about internetworking. Information isn't put into your computer so much as your computer is put into a world of information."[8]

You do not have to be a cultural anthropologist to suspect that "anytime-anyplace" universality is not the ultimate technological outcome. For technological reasons alone, we can see that fixed and specialized contexts will accumulate technology and diversify interactivity.[9] What is more, we are beginning to see how those contexts accumulate digital technology. Quite the opposite of a global brain (or Big Brother), arbitrary local aggregations of self-connecting systems can become islands of coherence in the chaos raised by pervasive computing. How to achieve this becomes more than ever a question of design.

	Virtual	*Physical*
Foreground	Graphical user interface (GUI)	Haptic interface
Background	Ambient interface	Inhabitable interface (smart space)

4.2 Beyond graphical interfaces

Many properties of ad hoc networking interest the designers of physical space. When connectivity lets a device discover what other appliances and services are on hand, then design necessarily involves processes that are best understood as local. The sheer volume of technological possibilities alone dictates that activity must be managed by context. Local properties of scale, discovery, protocol, configuration, and tuning all become essential. In comparison to current universal networks, local systems do not require such high-level models in software, and they are less subject to monitoring by external parties.

This emphasis on the local actually points away from Orwellian prospects. Alex Pentland, long a pioneering researcher in smart environments, has explained the advantages of locality by distinguishing "perceptual intelligence" from "ubiquity." "The goal of the perceptual intelligence approach is not to create computers with the logical powers envisioned in most AI research, or to have computers that are ubiquitous and networked, because most of the tasks we want performed do not seem to require complex reasoning or a god's-eye view of the situation. One can imagine, for instance, a well-trained dog controlling most of the functions we envision for future smart environments."[10]

Instead the intentionality must lie in human organizations, abilities, and practices. As we have seen, foundations in embodiment and activity theory help tilt the field of computer-human interaction (CHI) back toward understanding how people play situations. This new era in information technology is not about predictable outcomes so much as richer experience. It pursues much more human-centered goals in natural interaction.

We have seen how embodiment shapes expectations. From a philosophical standpoint, activity in context has supplanted tasks on isolated machines as the focus of information technology design. From architecture we have identified a latent need to map our embodiment onto the world. This pertains to the present discussion in that we feel a deep need to maintain technological constructs whose dimensions resemble those of the human body in architectural space.

Now let us apply this background to recent developments in technology. Physical devices establish possibilities for interaction beyond the desktop. Local models are necessary abstractions for technology-extensible places. Social situations provide design precedents and problems from which to build types. All of this points toward new forms of context-centered design. Whether the various goals are met, and whether this next layer of technology is to become a bane or a boon depends mostly on how many ways of environmental knowing can be brought to interaction design.

Understanding the Components

As computing moves beyond the desktop, what are the essential building blocks that nonspecialists may want to understand? As a way of reviewing recent developments, consider some essential categories in embedded computing technology. The elements and applications of ubiquitous computing are fairly well established within technical circles by now. Different emphases on their particulars distinguish research areas. Among those elements held in common, some should survive rapid changes in circumstance. Already these developments have consequence for the shift from virtual world building to physical computing.

Much of this work begins at the level of demonstrating technical possibility. Concerns for usefulness, organization, and social appropriateness inevitably follow. More so than engineered features, these situational factors cause some technologies to linger at the margin while others explode onto the global scene. It is notoriously difficult to predict how quickly developments will come on the market, and for what initial purposes. So put aside the technofutures for a moment. Here without speculation on their implications is a set as ten essential functions from which pervasive computing systems have been composed.[11]

1. Sites and devices are embedded with microprocessors.

It all starts with the embedded microprocessor. All sorts of things have a chip in them. The pocket radio of the 1960s, that first truly widespread instance of portable consumer electronics, was named for its embedded processor, which if not yet truly a chip, was at least "solid state." A beachgoer could listen to his or her "transistor."

By the year 2000, a mobile phone could pack more processing capacity and memory than a campus mainframe did in the 1960s. A web server could fit in a pocket. An ordinary chip could hold an operating system, a network interface, an Internet protocol stack, and a web client. In its extreme form, a web-capable device could be smaller and lighter than a nickel.[12] In 2001, news stories popularized knowledge of the "smart dust" developed at the University of California, Berkeley. Using solar power and optical transceivers, researchers overcame the usual size constraints presented by powered communications circuitry. At a 7-mm length, the prototypes were still too large to be blown about like actual dust, but for something able to communicate across the bay from San Francisco to Berkeley, these sensor devices were considered small.[13]

Technofuturists jumped on this miniaturization bandwagon with predictions of practical bionanotechnology within a decade. Nanotechnology takes embedded systems to the practical limits of smallness, with microns-wide devices that we would have difficulty understanding as chips. Biotechnology aims to integrate these with living systems. For present applications to architecture, portable gear, and temporarily worn devices, centimeter-sized devices are small enough.

The move beyond the desktop computer is well under way. According to U.S. government research statistics for the year 2000, shipments of new microcontrollers outnumbered those of new computational microprocessors by a factor of almost 50.[14] According to Intel, already more than 95 percent of devices containing microchips do not present themselves to their users as computers.

As demonstrated by surging enrollment in conferences such as those for embedded systems, device engineers have increasingly devoted their efforts to the interoperability of smaller chip-based devices in physical settings.[15] This technical interest occurs because the embedded device is engineered "low." Practical economies of engineering do not always warrant providing a full-service network operating system; devices can communicate at lower levels without that kind of overhead. Such internetworking is indeed vital. Without it, devices must be hard programmed for a particular purpose, like gadgets. With connectivity, however, embedded systems can communicate their status and receive ongoing instructions to and from their surroundings. In contrast to anytime-anyplace universality, this alternative is intermittent and local.

2. Sensors detect action.

If technologies are to keep out of the way, they need to see us coming. If computationally embedded environments are to be useful yet unobtrusive, they have to recognize what is happening in them. Next to the microprocessor itself, the next most basic component of embedded computing is the sensor. To the forecaster Paul Saffo, for instance, sensors have become the "key enabling technology" for computing. Microprocessors themselves led change in the 1980s, and laser optics (storage-rich compact disks and bandwidth-rich optical fibers) were key in the 1990s, "But we are beginning to see diminishing returns from merely adding more bandwidth to our access-oriented world. Now change is being driven by sensors—cheap, ubiquitous, high-performance sensors, or MEMS—and they will shape the coming decade."[16]

Like processors and networking before them, sensors have now reached the steepest part of the cost-reduction curve. Today a programmable network of wireless embedded sensors often costs less than one hard-wired dedicated single sensor circuit did not so long ago. Thus many more applications become normal. For example, a house may be equipped not only with smoke detectors, but also with inexpensive monitors for specific gases such as radon or carbon monoxide. On the streets of London, smog monitors mounted on lampposts transmit data to a nearby server. Devices under development there cost less than 1 percent as much as their predecessors.[17]

This invites a reconsideration of some basic engineering concepts. A sensor responds to a change in state. The medium in which the state exists might be mechanical, electrical, magnetic, hydrostatic, flowing, chemical, luminous, or logical. The change might be a discrete event, the gradual attainment of some threshold, or the establishment of a pattern. In effect, even the simplest mechanical sensor intrinsically serves a logic device, which simply reports whether a change has occurred. For example, a crude trip wire has two outputs: no, no one has walked by yet; or yes, someone just has.

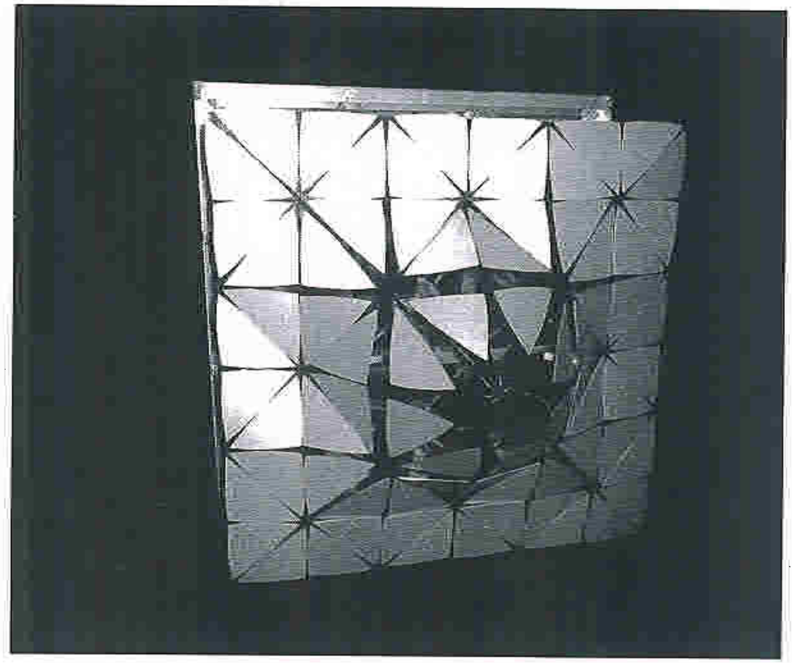
Typically in the history of engineering, some aspect of the physical configuration of a device, relative to its monitored medium, had to contain implicit information about what constituted a meaningful change. For example, a float valve responded to water reaching a prescribed level. Such a device would normally have to be calibrated to fit its setting, usually by means of some dimensional adjustment to a scale. Typically this kind of sensor would serve a single purpose, for which a sensing system would in effect be hard programmed as a mechanical computer. Optical and electronic systems have been more powerful, and many of these have long involved the use of microprocessors, but still they too have been operated in isolation and for a single purpose for which they have to be configured in advance.

Embedded computation changes that. Adaptable programmability is the key to the relationship between sensors and embedded microprocessors. Now the signal from a sensor can be interpreted statistically, over time, and in comparison with background conditions held in memory. This becomes especially powerful in large arrays. Systems for interpreting signals from a field of linked sensors have dramatically increased the capacity to recognize patterns. Wireless networking improves the practicality of distributed fields of sensors. Until recently, any mechanical-electrical sensor has required a full-time, direct, hardwired connection to its controller. Usually that has required a dedicated network, and at best it still had to share traffic on a full-service local area network. Often this connection has been more expensive than the sensor itself. Cost has limited the number and distribution of sensors, which in turn has limited the adaptability of whole systems. Now a field of wirelessly interlinked sensors can become practical. Interlinking can implement much more economical protocols for downtime, for passing or "hopping" messages directly among themselves, and even for reassigning tasks to one another. By analogy, such a field becomes more like a population, like birds that quickly pass the word when a cat is walking around beneath them. Continuous sensor fields can ask: "What trends are developing here, besides discrete events?"

The case of microelectromechanical systems circuits has been especially telling. According to estimates by an industry consortium, the MEMS market is expected to grow from \$2–5 billion in 2000 to \$8–15 billion in 2004. There were 1.5 MEMS devices per person in the United States in 2000, and that number was expected to grow at a compound rate of over 40 percent a year, to 5 such devices per person in the year 2004.[18]

The accelerometer, one common MEMS application, advances the cause of haptic interface design considerably. Instead of pushbutton clicks, interfaces employing a change of pace in continuous motion invite more skillful, embodied, and unobtrusive operations. The wave of a wand becomes much more of a reality. For example, one Nintendo GameBoy uses MEMS to allow video play by tilting rather than clicking. At larger scale, accelerometer-based gyroscopes replace spinning mass systems for navigating high-speed Acela and TGV trains.[19]

Also relevant to embodied interaction design, pressure sensing has become more practical. A pressure-sensitive resistor now costs little enough that it can be used in casual applications. When distributed in an array, these can make a building surface responsive to human presence (figure 4.3). In a demonstration of this possibility, Hewlett-Packard's David Cliff has rigged a pressure-sensitive dance floor



4.3 A responsive building surface: Aegis Hyposurface. (*Courtesy of Mark Goulthorpe, dECOi.*)

through which the activity of dancers is fed back into a musical selection or real-time composition. If it were coupled with wearable biofeedback devices such as heart-rate monitors, or motion accelerometers, and with breeder algorithms in the composition software, this could provide a fresh trance scene without recourse to chemicals.[20]

Although the range of physical possibilities expands, nevertheless most sensing tends to remain visual. Computer vision has been intrinsic to many agendas in pervasive computing. This is because identifying the actors is such a fundamental issue in context-aware software and because tagging is not always welcome, practical, or sufficient. Abilities in vision are relative, of course. An assemblyline robotic arm can "see" the objects it needs to pick up, but nothing more. Almost as quickly as the state of sensing advances, however, old fears about panoptic social control reawaken. Cameras appear in more places all the time, and we are rightly afraid of them.

According to ctrl [SPACE], an exhibition and publication from art and media center Zentrum für Kunst und Medientechnologie (ZKM) in Karlsruhe, Germany, it is not only visual sensing by cameras, but tracking by various forms of what the organizers called "data-veillance" that should concern us.

Sensors are not only the most essential but also the most unnerving of the new technology elements. Concerns about privacy make most of us unreceptive to this technology. But that is a larger social question. As Scott McNealy, chief executive officer of Sun Microsystems, said in 1999: "You have zero privacy anyway—get over it."[21] We can only hope that with billions of sensors in place, there is too much information for any software, much less some government agency that cannot even master its own internal communications, to interpret successfully.

3. Communication links form ad hoc networks of devices.

Local-hop sensor fields are just one example of how networks may temporarily form. Instead of being intensively planned and rigged, pervasive computing depends on unplanned communication. To avoid technological overload and privacy losses, connections are opened only where necessary.

Practical links can be fixed or portable, specialized or general, constant or intermittent, and passive or interactive. Their usability depends on interrelationships as in an ecology. Patterns of unplanned communication may allow unanticipated local capacities to emerge.

Even among purely nomadic technologies, context has begun to play a role. The radio-based Bluetooth protocol, introduced in the late 1990s, quickly became a standard for linking nearby devices. Five years later, a universal serial bus (USB) chip could be used to make peer-to-peer connections between devices formerly requiring a host computer such as a desktop PC, and at a bandwidth considerably higher than wireless Bluetooth. Not all linked objects will benefit from a full-featured web browser, of course. More will run some slimmer set of communications.

Thus when current developments start from the idea of a web connection, they often work toward much "lighter" (lower communications overhead) and more flexible connections. Web protocols as we know them do provide a convenient way to use interactive computers as we know them. They let us monitor embedded computers and sensors in applications where the networking cost is justified. Server-side computations can employ powerful computers and high-level programming languages to manage smaller devices that are not so easily codable in themselves. Meanwhile, some web servers can now fit in one's hand—or in a hand-sized object.[22]

But smaller objects are codable—that is the whole point of the hype. As has been fairly common for some years now, the Java model lets programs developed at a high level be executed locally on small machines. Most microprocessors have the power to serve as a Java "virtual machine"; such processing logic is not the main engineering constraint (power supply, for instance, can be more the issue). Moreover, devices running Java software can poll their vicinity to find out what other relevant devices are available for interconnection. In 2000, Sun Microsystems' JINI standard introduced a "network dial tone" by which such polling could occur economically. The increased connectivity that results from such a discovery service increases the likelihood of a division of labor. For example, a small Java device that needs some interactivity but cannot justify buttons or a display can offload its interface to a larger device in the vicinity. A larger device monitoring the activity of several smaller ones can invoke their local computations remotely. Furthermore it can download different software onto those devices to change their role. Networked communication thus dramatically increases the capacity of a local collection of devices to adapt to incidental conditions.[23]

Ad hoc communication thus adds new capacity for the appropriation of context. Fixed resources may suggest, and to some extent become adapted by, different configurations of software. This is fundamentally different from an anytime-anyplace uniformity, and it is an important property of place. Adaptive reassignment of locally networked devices raises the prospect that pervasive computing constitutes a new type of recyclable resource. Dan Siewiorek, director of the Human Computer Interaction Institute at Carnegie Mellon, has called this "renewable" computing. This occurs not only at the level of local networks, but a larger and smaller scales as well. "The micro level considers harvesting energy from the surroundings. The infrastructure level addresses renewable structures and how services emerge as the mix of devices evolves. The macro level considers the transformation of information sharing and impact on existing systems such as transportation and industry."[24]

Renewable, location-based infrastructures constitute quite a different future from that of disembodied cyberspace; yet the present state of pervasive computing is neither of those. Communication between interactive portable devices dominates current developments. These are still in the foreground; they are not yet much of a periphery.

As with any technofuture, there is a big difference between the newly possible and the generally practical. Intel director of research David Tennenhouse has cautioned: "Radical innovation will be required to bring networking costs in line with the $1-per-device price structure of the embedded computing market."[25]

4. Tags identify actors.

Contextual awareness begins from an ability to recognize who or what is present. Pattern-recognizing sensors can detect some of this, but for habitual applications, or for ambiguous conditions, recognition becomes much simpler when something is simply tagged. This is forecast to grow into a billion dollar industry in America in this decade. A radiofrequency identification tag (RFID) cost less than a dollar in 2001, and is expected to cost less than a penny with a few years.

Tagging based on earlier information technologies is already widespread—especially the bar code. As a great many news stories proclaimed, 1999 marked the quarter century anniversary of the uniform product code (UPC), which is the standard that brought widespread success to optical tagging, first of groceries, (although the first items imprinted with bar codes were railroad cars), then with most retail goods, and sometimes even people themselves, such as hospital patients. The electronic product code (EPC) that has been designed to replace the UPC takes this further by giving a unique number not only to each product, but to each unit of that product.

Tagging has abundant architectural precedents. Ornament, inscriptions, and signage have all in effect tagged the built environment with information intended for particular purposes. In some cases, especially where automobiles dominate traffic, signage has become more prominent (and often more invested with design) than the buildings themselves. Tagging now complements the signs and symbols that architectural theorists know so well with more diverse forms and smaller scales than has been practical before. Some of these are quite pervasive, such as tagging places by means of their coordinates in a global positioning system (GPS).

The addition of computation to tagging yields a lot more applications besides pricing and gatekeeping. Just about any barcode application is coupled to an inventory control system. Smart touch-sensitive counters, such as those that have been demonstrated on bins in parts warehouses, can conveniently monitor the flow of goods outside all the usual point-of-sale documentation.[26] Any place that large numbers of such tags are read, particularly the retail point of sale, in effect becomes an important component in the use of space. The overall flow in a system of tagged items might only be visualized by analysts at some central hub. Major retailers such as Walmart have thus been among the first to pilot use of RFID tags on a large scale for tracking low-cost merchandise as it moves off the "smart shelves."

A tag can be used to summon annotation as well. For example, Sony has demonstrated a system for summoning biographical and professional data on office occupants by means of tags from bar codes on office doors (the Navicam).

Tagged items can function as physical tokens in hybrid physical-digital systems. In essence, a token represents an abstract arrangement with a physical symbol. This is not to say that we want to embed a text of vows into a wedding ring; tagged tokens seem to take the more mundane form of yellow stickies and refrigerator magnets.

For example, in "Triangles," one of the first broadly recognized projects in the Things that Think consortium at the MIT Media Laboratory, palm-sized physical triangles served as tags for project data; attaching them to one another or waving them in front of a camera computer triggered operations on a desktop computer. Early on, Hiroshi Ishii's group demonstrated the principle of using tagged objects as "phicons" (physical icons) or "metablocks" as interface widgets.[27] One obvious connection is to link physical tokens to websites. Even a banana at the supermarket has a web address printed on its price code sticker.

From a technological standpoint, smart tagging brings software into the physical environment by means of small, affordable Java virtual machines. For example, "iButton" is a 16-mm portable Java platform that can be affixed to a keychain, a ring, a wall, or the side of a computer. As of 2002, more than 65 million iButtons were in circulation, and one could order them in lots of ten thousands on the Internet.[28] As buttons, one could sew these onto a shirt, or make smart jewelry from them. As an inescapable part of embodied computation, computation moves onto the body.

Tagging *people* raises a lot more questions. We all carry identification cards with magnetic stripes. A Harvard faculty ID will cease to let you into your old building less than a month after your appointment has ended. In early 2002, Hong Kong introduced smart ID cards that included biometric information such as a thumbprint, as well as a picture, birth date, and address.[29] However, a card is still vaguely private in that you keep it in your wallet.

"Biometric" identification has been made topical by recent interest in domestic security. It involves no stealable passwords or tokens. One of the first "biometric" authentication devices to come to market at an affordable price ($200) was offered by Ethentica.com. In 2001 the company began offering digital fingerprint authentication; owners could attach fingerprint security to individual data and applications, as a substitute for remembering and typing.

Tagging becomes more blatant wherever people agree to wear badges. In a world full of mobile workers, badges appear to have become an accepted fact of life, at least in workplaces. Social scenes have their own subtle badges of standing too, of course, and when these start polling and linking it can make for quite a scene indeed.

A badge that functions within a limited radius, that allows passage of a particular perimeter, and which (occasionally) helps customize local resources for its holder has not surprisingly been among

the earliest testbeds for pervasive computing. A smart badge brings all the technical, and more important, all the social issues into focus. Badges were intrinsic to Xerox PARC's early work on pervasive computing. These were inherently contextual as well; they functioned in specified locations only, in confederation with other digital devices at larger scales. Similarly, the Olivetti Smart Badge, often cited as an earliest precedent in this technology research, was a site-based, context-aware application.

5. Actuators close the loop.

In a sense, whenever a system regulates itself by monitoring its own performance (i.e., with a feedback loop), some rudimentary intelligence is implied. By this reasoning, even an ancient water clock was "smart." Similarly, a household thermostat, based on thermal expansion of a copper coil to meet an adjustable electrical contact, is an analog computer of a sort. As a switch it is an actuator—a device that alters a system's state when it is triggered by appropriate conditions.

Modern industrial process engineering has been based on devices, such as the servomotor, that translate electronic signals into physical actions. Digital systems expand the linkage and logic behind these control signals. Besides sensors, then, embedded gear advances and diversifies the role of actuators.

Technofutures have been rife with these. Wherever industrial processes and products have employed feedback control systems, popular imagination has extended their application to domestic, social, and recreational uses. Today the old Jetsons fantasy about mechanical devices that pamper us (or the much more mod-cybernetic version of same from the magazine *Archigram*) has been renewed with digital devices that adapt to us. Neo-Jetson enthusiasm for the smart appliance revives all the questions of how to keep actuators in the periphery.

We want environmental systems to keep out of the way, but we also want them to do something. We might be interested to see "environmental controls" do something more interesting than just stabilize indoor climate. This is not all fantasy; many of these exist. For half a century, Disney imagineers have excelled at putting actuators into small objects, for example. An "animatronic" theme park character contains dozens of servomotors. A recent luxury car is also loaded with actuators, and many of them are coupled to fairly sophisticated computers. Instant airbags and antilock brakes receive most of the press, but fuel systems, valve systems, steering systems, vibration isolation, suspension, even seat adjustments employ meters, timers, gates, and especially a lot of fast, small actuators to improve whole-system response.[30]

In architecture, applications began with resource management systems, particularly those for energy, which especially in America have seemed anything but smart. Computer-driven actuators can adjust sunshades, schedule peak demand loads, manage the much more complex states that result from much more localized control, and so on. Maintenance systems have spread as well; sensors built into structures can identify deterioration and signal for upkeep before failures occur. Bridges and dams, whose failure would be catastrophic, increasingly use this technology.

Building geometry itself has sometimes seemed suitable for kinetic systems. Embedded kinetic elements manipulate other components of buildings, like tendons moving skeletons. Deployable kinetic elements are temporary structures, like tents. Dynamic elements are independently mobile, but are affixed to buildings, like doors and canopies.[31] For example, responsive systems have received some press in Enrique Norten's design for the Educare school gymnasium in Guadalajara, Mexico, in which sensor-controlled enameled steel panels open and close, in the words of the architect, "like the scales of a fish, or the feathers of a bird."[32] In the prominent first American building by Santiago Calatrava, a pavilion for the Milwaukee Art Museum, the 200-foot sunshade over the reception area opens "like the wings of a bird."[33] These sculptural applications have moved sensing into the profession's imagination in a way that other climate-control systems cannot. And by monitoring atmospheric rather than social conditions, they have done so in a way that is not alarming.

The physical environment abounds with opportunities for improving commodity, firmness, and delight through the application of intelligent feedback systems. Commodity is largely a matter of life-

cycle economy. Energy, materials, and space are wasted at prodigious rates in built environments.[34] Firmness invites applications for systems that provide safety and security in a more intelligent, less obtrusive way than today's building codes. Meanwhile, delight is up for grabs. Places that afford some participatory adjustment on daily and seasonal cycles have long been alleged to be more satisfying than those that are uniform. Spaces that subtly reconfigure themselves according to their occupants and use can cause paranoia or delight, depending on how intelligently they are designed.

6. Controls make it participatory.

If all this technology were completely automatic and able to function completely passively, it would be out of the way all right—and more frightening than ever. Instead, smart systems need to be operable where it is appropriate. It might be preferable to configure some systems just once, to adjust others occasionally, and to incorporate a few into daily routines. Like a remote control for television, some might be twiddled for amusement. Like a musical instrument, some smart systems can facilitate personal growth through skilled practice.[35] These are basic principles of interaction design: Know when to eliminate an obsolete "legacy" operation, when to automate, and when to assist an action. Know how to empower, not overwhelm.

"Pushbutton" convenience was a hallmark of a modern age only recently freed from bodily work. The industrial button was once understood as an abstraction. The automated operations it triggered were still fresh substitutes for something much more tedious. The postindustrial button frequently lacks a referent in bodily experience, however. This is partly because its logic leads to cumbersome convolution. Hundreds of buttons, or hundreds of expressions for entry by buttons, depart from the realm of comprehensible experience. This is why the household videocassette recorder has become a standard emblem of incomprehensibility.

Haptic interface strategies based on gestures, gliders, and motion sensing provide alternatives to the current excess of clickable buttons.[36] For example, the tilt-and roll technologies of the MEMS-equipped Gameboy can be carried over to the a handheld computer.

This may be used for gaming at first, but once people are comfortable with the operation, it can be used in other applications. Motion Sense plug-ins extended capacity this to Palm devices in 2001, for example.

Where the sensitivity of applications warrants investment in much more specialized interfaces, haptics technology has advanced considerably. For example, the ReachIn interface, used in training persons for medical procedures, combines the Phantom pointing device with a reflected transparent screen that puts a virtual display right over the active hand. With a repertoire of specialty software for moving and modifying surfaces, and for rendering texture and friction, this arrangement has delivered the best multimodal haptic interface to date. "Haptic rendering," the process by which virtual surface properties are communicated through an ultrasensitive force-feedback device in real time, bears many analogies to the graphics rendering of a decade earlier. Like graphics, these algorithms eventually end up in hardware, on a chip, where they become practical for more casual application, such as in everyday geometric modeling in computer-aided design.[37]

We know from traditional craft that finesse requires some touch but not necessarily full embodiment. This is an important distinction. The usual objection has been that symbolic processing requires us to sit still. Relative to traditional work practices, this was more or less true. But as we have discovered from the first few decades of creative computing, the active participation so conducive to learning and expertise wants more than a mouse and a screen. Even within the crude window-icon-menu-pointer (WIMP) technology, there is enough affordance to support talent. The way is wide open toward physical interaction design: active controls are integrated with comprehensible, satisfying things.

7. Display spreads out.

For a precedent in ubiquitous information technology, Mark Weiser would point out text. Text really is ubiquitous—you are rarely out of sight of several pieces of it. Take cars; as anyone who has time to study them amid gridlock on the freeway knows, the back end of a car can signal you with at least five text sites: the bumper sticker, the license plate frame, the characters of the plate itself, the dangling sign cautioning about whomever is on board, and the window decal.[38] The back of a cereal box is a cacophony of texts and images vying for the awakening person's momentary attention. And, in an act that would most likely astonish a visitor from any other century, millions of people think nothing of going out wearing unpaid advertising in the form of logos and messages on their clothes.

This is a tale worth its many recitations. Once upon a time text was scarce, and was generally confined to the library. Occasionally some of it would be chiseled onto buildings, but that too was a rarefied setting. Modern printing obviously changed all that. "This will kill that," said Victor Hugo of the printed word's advantage over the inscribed building. In the past 50 years, printing and photo composition have moved text and images into unprecedented contexts. Next, text displays came alive. The word processor is one reason why most of us tolerate computers. More recently the scale of displays has expanded down to handheld devices and up to cover whole walls. As evidence that information truly has become ubiquitous, the text screen on a gas pump scrolls an advertisement while you are filling your tank.

The idea that how we do things with symbols depends on their scale and position relative to the body is fundamental to pervasive computing. Something you read inclined on the sofa should be a different size than something you read as you step out of sidewalk traffic into a doorway. At 10 feet high an image reads very differently than the same content at 5 inches. Scale was intrinsic to the original notion of ubiquitous computing: Weiser's group at Xerox PARC made a typological distinction among tabs, pads, and boards.

Lately it has become possible to move text between many scales and surfaces. Researchers from Sony have demonstrated ways to drag an image off a laptop computer and onto a wall.[39] The greater the variety of display surfaces on hand, the more appealing such mobility becomes. For an example of free-form display, IBM's Everywhere Displays project combined projection with detection on an arbitrary surface such as a tabletop or a wall.[40] In effect, this coupling turns the surface into a crude wireless touch screen.

The expression *augmented reality* has generally been reserved for conditions in which a virtual display is overlaid onto a physical scene. Sometimes this occurs on the eyeglasses of a viewer. Such applications have been pioneered on assembly lines and in equipment repair, for example, where annotation of viewed objects is needed but the hands have to be kept free.

Nevertheless it is the live display in the fixed setting that may most characterize embedded systems, and that brings pervasive computing into architecture. Again the earliest examples were in the laboratories. At MIT, Alex Pentland's ALIVE project team maintained a wall-size screen suitable for animated characters.[41] Hiroshi Ishii's "ambient ceiling" set an early standard of being in the periphery.[42] It was based on projected images of ripples like those on the surface of a pond, whose frequency and intensity were mapped to local measures of ambient conditions such as network traffic or the number of people on site. Wall-sized displays have been important to more recent interactivity research at Stanford University. These "murals" provide much more substantial content than previous projected displays, while keeping the control elements to a minimum. Laboratory director Terry Winograd has often used architecture metaphors to describe how, in the design of interactive experience, the whole is greater than the sum of the individual technical components.[43]

Many people think of large-format displays in terms of the year the NASDAQ sign went up. This outdoor installation demonstrated that besides size, another factor important to ubiquitous display is its ruggedness. A display that can be left out in the rain opens a very different realm of imagination.

Next, the literal ground itself becomes interactive. With the spread of positioning systems, which in effect make anyone who carries such a system into a live cursor, the city plan itself becomes a living surface.

Perhaps because we have been culturally conditioned to fantasies being visual, or because the dominant cultural media of the last century would have been wild fantasies in the one before that, prospects in display technology make it more difficult to avoid technofuturism.

8. Fixed locations track mobile positions.

Positioning technology has exploded. In a 2002 interview, GPS pioneer Per Enge observed that in contrast to original projections for an eventual market for 40,000 GPS units (mostly in military applications), at this point 100,000 were being produced each month.[44]

"Let's put GPS in necklaces and dog collars. Everything that moves should have GPS," wrote Kanwar Chadha, chief executive officer of SiRF, the leading GPS chipset manufacturer.[45] "This kind of stuff has enormous potential for abuse by the authorities, or by anyone who can break into the information," wrote Emily Whitfield, a spokesperson for the American Civil Liberties Union.[46] Concerns include not only the usual fears of governments monitoring their citizens, but also criminals tracking their prey.

Among the ways to achieve more natural interaction design, there is none quite so obvious as using position for input. You do not have to be a genius to understand the potential of a device whose sole instructions are to take it somewhere and turn it on. Maybe a third step would be to pop in a filter for what you want to know: plant identification, history of a city, bar hopping, tracking your friends, finding your tribe.[47]

Because space is such a fundamental category, position can serve as an index to a range of information services. Any body of data that can be "geocoded," that is, assigned a position as a key to a record in a relational database, can be delivered intelligently through geographic information systems (GIS).[48] With mobile communications, the information can be delivered where needed. A GPS completes the loop; position data are used to query huge spatial databases that report relevant information back to the position being described. That information can be highly thematic, and not just the stuff that usually shows up on printed roadmaps. For example one might use a mobile GPS-GIS system to study vegetation patterns, ethnic neighborhood boundaries, or current nightclub scenes.

When coupled with tagging, positioning technology helps track things that move around. This helps answer such fundamental questions as "Who is here?" and "What are they doing?" Like the products in a retail supply chain, elements of other networked distributions become documented and their flows become better modeled. Knowing where things are leads to natural economies of routing. Even a sidewalk vendor knows there is value added in this. Tracking also improves the most rudimentary aspects of sport, as any hunter knows. Tracking distributed fields of movements raises new prospects in scientific and recreational visualization. Social delights and abuses quickly occur to the technofuturist imagination. When it is combined with architecture's role of arranging bodies in space, for example, tracking seems more like a security application than play.

Positioning systems also cater to the geographically unskilled. In the late 1990s, Hertz began offering "NeverLost" service automobile navigation system in certain rental areas. "Whereify," a wristband device for tracking a child, was one of the first commercial GPS wearables (figure 4.4). This was just the tip of the iceberg on intelligent transportation systems (ITS), which in turn were just one sector of what was already a multibillion dollar market in geodata.

4.4 Wherify GPS Locator, a location-tracking wearable for children. (*Courtesy of wherify.com.*)

With the cost of a GPS chipset falling through $10, and with the repeal of its resolution reduction for nonmilitary applications as of April 2000, the way for computers to annotate the physical world was been cast wide open. "Urban markup language" does not yet exist.[49] However, in 2001, a geography markup language standard was introduced.[50]

9. Software models situations.

As hardware becomes less expensive, more diverse, and more plentiful, software becomes more challenging. Representing scenes and situations becomes the essential challenge. Knowledge representation remains perhaps the fundamental challenge in software. As evidence for this, the discourse has shifted from artificial intelligence to ontology; that is, to representing the existence of people, actors, and things and their contexts. "Who is here and what are they doing?" recalls Laurel's foundational notion that the purpose of computers is to let people take part in shared representations of action.[51] Add the many components of pervasive computing to the means, and the pursuit of these ends becomes more interesting. For example, a system may begin to model a physically proximate area by polling local ad hoc links between known mobile tags and devices. Protocols, etiquettes, and other such aspects of social framing become more essential than they were in earlier, desktop realms of computing. Architectural frames, and—despite exaggerated reports of its death—environmental scale and geometry help organize so much information.

Location models appear so critical to the problems of implementing contextual awareness that they deserve much more discussion in this review. Chapter 5 explores the relations between geometry, the geodata industry, and sensor-actuator systems, all in relation to the problem of representing actors on stages. Chapter 6 assembles one possible set of situational types for which such models may serve. Persistent goals in knowledge representation carry over from the world of ambitious desktop artificial intelligence to less versatile, but more numerous instances of information appliances, smart spaces, and interactive environmental controls. For example, the metaphor of the butler still obtains. Like a butler or personal secretary, some contextual computing must anticipate some of our needs before we do, and must carry out some of our business without our needing to know about it; but it must maintain our identity, accessibility, and etiquette where appropriate in the processes.

10. Tuning overcomes rigidity.

Even before any software location models are implemented, even the crudest aggregation of hardware and links must be lived in, lived with, and tuned. Much of the place-centered character of situated interaction design comes from the fact that any fixed collection of devices has to be integrated. Tuning consists of incremental adaptations of configurations and settings based on a qualitative, top level interpretation of the performance, and in best cases, the "feel" of the aggregate. Acousticians tune concert halls. Game developers tune the variables and constraints in strategic simulations. Even when engineers balance complex systems using mathematical models, some tuning creeps in. The prevalence of tuning in today's culture of technology usage is demonstrated by the spread of the word *tweak*.

Tuning includes incremental growth and change. How are new devices added? What model underlies the world in which all of these interoperate? Must the whole system be rebalanced each time it incorporates another element? Steve Shafer from Microsoft Research has argued that not only is tuning necessary, but indeed it becomes one of the central knowledge representation problems of the emerging generation of research.

> The most obvious thing about ubiquitous computing or intelligent environments is that they have a lot of devices talking to each other. Accordingly, one of the first questions in building such systems is how to get these things to talk to each other at all. This raises questions of network protocols, distributed object programming systems, etc, and much research proceeds along these lines, as though that were sufficient for ubiquitous computing. But, assuming the connections are made somehow, the deeper question is, how can we make these interactions meaningful? What would it take to call device interactions meaningful?[52]

Then the systems start to tweak themselves. In contrast to a sense of place, consider places with sense. Smart spaces recognize at least something about what is going on in them, and then they respond.

Some of this built-in understanding now can reside in easily adaptable software, some can be implicit in occasionally reconfigurable arrangements of furniturelike hardware, and some remains better off being built in. It is the interrelationship of these that needs design.

The question exists of whether each smart space must be built in an ad hoc manner, or whether standards and reusable software components will emerge. Such an infrastructure would be somewhat analogous to the "event handler" typical of interactive desktop windowing and multimedia scripting systems, but would be infinitely more complicated in the continuous context of a physical space. Multiple actors, ambiguous aggregations of objects, and unreliable data streams from distibuted sensor fields all replace the mouse click as inputs to be interpreted. Researchers in this must extrapolate from what they know. Programmers from the AwareHome project at Georgia Institute of Technology have developed "context widgets," for example. These are analogous to desktop widgets, reusable modules with fixed sets of callbacks to other elements of the system.[53]

From a systems engineering standpoint, tuning is a matter of regulating the transfers between component devices to achieve an overall system performance that is not easily predicted by numerical methods. Quantitative analysis seeks optimization, which it tries to predict with indicated solutions to mathematical models. Few design problems afford such determinacy, however. Under- and overconstrained problems produce a complexity where human judgment is better than predictive formulations. Here the approach to tuning, and to the design philosophy it represents, is more propositional. This may sound exactly like the "try it and see" approach to design improvization that decades of numerical analysis have sought to overcome, but more accurately it illustrates the need for a partnership between prediction and invention in creative problem solving.

Toward a Typology of Situated Interaction Design

If the tuning of smart environments were to rely exclusively on ad hoc inventiveness, work would proceed slowly. If it were to reduce its considerations to functions that could be modeled more predictably, this effort would produce sterile results. In between these approaches, something is needed in the way of continuous, if not fully formalized knowledge. Invention needs to play off convention. Unless engineers are to face a debilitating agglomeration of gear, some aspect of context has to help with tuning and protocols.

That aspect is type. Persistent structures of form and environment should be able to accomplish half the work of tuning aggregations of portable and embedded technology. If, for example, one is tuning smart gear for a café, a lot of the work should be accomplished by the fact that this is a café. Location and type have to matter. Otherwise, with everything possible all the time, mostly chaos will result.