Connecting the Physical World with Pervasive Networks

This article addresses the challenges and opportunities of instrumenting the physical world with pervasive networks of sensor-rich, embedded computation. The authors present a taxonomy of emerging systems and outline the enabling technological developments.

Deborah Estrin
University of California, Los Angeles

David Culler and Kris Pister
University of California, Berkeley

Gaurav Sukhatme
University of Southern California

Mark Weiser envisioned a world in which computing is so pervasive that everyday devices can sense their relationship to us and to each other. They could, thereby, respond so appropriately to our actions that the computing aspects would fade into the background. Underlying this vision is the assumption that sensing a broad set of physical phenomena, rather than just data input, will become a common aspect of small, embedded computers and that these devices will communicate with each other (as well as to some more powerful infrastructure) to organize and coordinate their actions.

Recall the story of Sal in Weiser's article; Sal looked out her window and saw "tracks" as evidence of her neighbors' morning strolls. What sort of system did this seemingly simple functionality imply? Certainly Weiser did not envision ubiquitous cameras placed throughout the neighborhood. Such a solution would be far too heavy for the application's relatively casual nature as well as quite invasive with respect to personal privacy. Instead, Weiser posited the existence of far less intrusive instrumentation in neighborhood spaces—perhaps smart paving stones that could detect local activity and indicate the walker's direction based on exchanges between neighboring nodes. As we have marched technology forward, we are now in a position to translate this aspect of Weiser's vision to reality and apply it to a wide range of important applications, both computing and social.

Other articles in this issue address the user interface-, application-, software-, and device-level design challenges associated with realizing Weiser's vision. Here, we address the challenges and opportunities of instrumenting the physical world with pervasive networks of sensor-rich, embedded computation. Such systems fulfill two of Weiser's key objectives—*ubiquity*, by injecting computation into the physical world with high spatial density, and *invisibility*, by having the nodes and collectives of nodes operate autonomously. Of particular importance to the technical community is making such pervasive computing itself pervasive. We need reusable building blocks that can help us move away from the specialized instrumentation of each particular environment and move toward building reusable techniques for sensing, computing, and manipulating the physical world.

The physical world presents an incredibly rich set of input modalities, including acoustics, image, motion, vibration, heat, light, moisture, pressure, ultrasound, radio, magnetic, and many more exotic modes. Traditionally, sensing and manipulating the

1536-1268/02/$17.00 © 2002 IEEE

physical world meant deploying and placing a highly engineered collection of instruments to obtain particular inputs and reporting the data over specialized wired control protocols to data acquisition computers. Ubiquitous computing testbeds have retained much of this engineered data acquisition style, although we use them to observe a variety of unstructured phenomena (such as human gestures and interaction). The opportunity ahead lies in the ability to easily deploy flexible sensing, computation, and actuation capabilities into our physical environments such that the devices themselves are general-purpose and can organize and adapt to support several application types.[1]

In this article, we describe the challenges on the road ahead, present a taxonomy of system types that we expect to emerge during this next decade of research and development, and summarize technological developments.

Challenges

The most serious impediments to pervasive computing's advances are systems challenges. The immense amount of distributed system elements, limited physical access to them, and this regime's extreme environmental dynamics, when considered together, imply that we must fundamentally reexamine familiar layers of abstraction and the kinds of hardware acceleration employed—even our algorithmic techniques.

Immense scale

A vast number of small devices will comprise these systems. To achieve dense instrumentation of complex physical systems, these devices must scale down to extremely small volume, with applications formulated in terms of immense numbers of them. In five to 10 years, complete systems with computing, storage, communication, sensing, and energy storage could be as small as a cubic millimeter, but each aspect's capacity will be limited. Fidelity and availability will come from the quantity of partially redundant measurements and their correlation, not the individual components' quality and precision.

Limited access

Many devices will be embedded in the environment in places that are inaccessible or expensive to connect with wires, making the individual system elements largely untethered, unattended, and resource constrained. Much communication will be wireless, and nodes will have to rely on onboard and harvested energy (such as from batteries and solar cells). Inaccessibility, as well as sheer scale, implies that they must operate without human attendance; each piece is such a small part of the whole that nobody can reasonably lay hands on all of them. At sufficient levels of efficiency, energy harvested from the environment can potentially allow arbitrary lifetimes, but the available energy bounds the amount of activity permitted per unit time. Energy constraints also limit the application space considerably; if solar power is used, nodes must be outdoors, and if batteries cannot be recharged, they will seriously affect maintenance, pollution, and replacement costs.

Extreme dynamics

By virtue of nodes and the system as a whole being closely tied to the ever-changing physical world, these systems will experience extreme dynamics. By design, they can sense their environment to provide inputs to higher-level tasks, and environmental changes directly affect their performance. In particular, environmental factors dramatically influence propagation characteristics of low-power radio frequency (RF) and can effectively create mobility even in stationary configurations. These devices also experience extreme variation in demand: Most of the time, they observe that no relevant change has occurred and no relevant information has been communicated. Thus, they must maintain vigilance while consuming almost no power.

However, when important events do occur, a great deal happens at once. Flows of high- and low-level data among sensors and actuators must be efficiently interleaved while meeting real-time demands, and redundant flows must be managed effectively. Consequently, passive vigilance is punctuated by bursts of concurrency-intensive operation. To cope with resource limitations in the presence of such dynamics, these systems will be governed by internal control loops in which components continuously adapt their individual and joint behavior to resource and stimulus availability.

A taxonomy of systems

Meeting these challenges requires new frameworks for system design of the sort that only come from direct, hands-on experience with the emerging technological regime. We must apply many small, power-constrained, connected devices to real problems where programming, orchestration, and management of individual devices are impractical.

The applications of physically embedded networks are as varied as the physical environments in which we live and work. Yet, even with this heterogeneity, many opportunities and resources for exploiting commonality across them exist. Most important is the hope to achieve pervasiveness by enabling system reuse and evolution. Within any one environment or system, more than one type of system or application might be active as the system evolves or from its outset. One step toward identifying common building blocks is to define a taxonomy of systems and applications so that we can identify and foster reusable and paramaterizable features.

We divide the space of physically embedded systems along the critical dimensions of space and time. Within these dimensions, we are interested in *scale*, *variability*, and *autonomy*. We address these dimensions with respect to the environmental stimuli being captured and the system elements.

Scale

Spatial and temporal scale concerns the sampling interval, the extent of overall system coverage, and the relative number of sensor nodes to input stimuli. The scale of spatial and temporal sampling and extent are important determinants of system emphasis. The finer grain the sampling, the more important the innovative collabora-

tive signal processing techniques such as those described elsewhere.[2–5] However, systems intended to operate over extended periods of time and regions of space must emphasize techniques for self-organization because as the system extent grows, the ability to configure and control the environment becomes infeasible. High density provides many opportunities and challenges for self-configuration that low-density systems don't encounter.

Sampling. The physical phenomena measured ultimately dictates spatial and temporal sampling scale. High-frequency waves require higher temporal and spatial sampling than phenomena such as temperature or light in a room, where spatial and temporal fluctuations are coarser-grained. The sampling scale is a function of both the phenomena and the application. If you need the system for event detection, the requirement could be more lax than if the goal is event or signal reconstruction. For example, if a structure is being monitored to detect structural faults, signatures of seismic response might be compared to "healthy" signatures at a relatively coarse grain. However, if data is being collected to generate profiles of structure response, fine-grain data is needed.

Extent. The spatial and temporal extent of systems also varies widely. At the high end of this continuum are environmental monitoring systems, which can span on the order of tens of thousands of meters. Most existing and planned pervasive computing systems are an order of magnitude, or smaller, such as is needed to cover a building or room. Elements of pervasive computing systems also extend to the smaller end of the continuum such as reconfigurable fabric, which a user can wear or that we can deploy to monitor structure or machinery surfaces.

Density. System density is a measure of sensor nodes per footprint of input stimuli. Higher-density systems provide greater opportunities for exploiting redundancy to eliminate noise and extend system lifetime. The higher the density of nodes to stimuli, the greater the number of independent measurements possible and thus opportunities to combine measurements to eliminate channel or coupling noise. Similarly, where density is high enough to allow oversampling, nodes can go to sleep for long periods of time and thereby extend coverage over time.

Variability

Variability is a second differentiating characteristic of many systems and associated designs. As with spatial and temporal scale, it takes on many forms and can apply to system elements or the phenomena being sensed. Relatively static systems emphasize design time optimization whereas more variable systems must use runtime self-organization and might be fundamentally limited in the extent to which they are both variable and long-lived.

Structure. Ad hoc versus engineered system structure refers to the variability in system composition.[6] At one extreme is a system that monitors a structure such as a building, bridge, or even an individual airplane. At the other end are sensor networks deployed in remote regions to study biocomplexity. More traditional pervasive computing systems embody aspects of both systems; elements of the systems such as instrumentation in an aware room or house might be relatively static, whereas the larger system that includes humans and subsystems carried in and out of the space is clearly ad hoc.

Task. Variability in system task determines the extent to which we can optimize the system for a single mode of operation. Even a structurally static system might perform different tasks over time—for example, a structural system that periodically generates a structure profile could act as an event monitoring system. Similarly, a system used regularly to measure an air conditioning system's effectiveness might occasionally have to integrate inputs from new sensors to detect traces of newly characterized toxins

Space. Variability in space—meaning mobility—applies to both system nodes and phenomena. In many systems of interest, most or all the nodes remain fixed in space once placed. However, many interesting, if longer-term, systems include elements that move themselves or that are tied to objects that move them (such as vehicles or people). Similarly, the phenomena these systems monitor differ in the extent of mobility. Many systems of interest are intended for phenomena that move quickly in time. A system designed to manage the physical environment for a stationary human user faces different challenges than one designed to track humans moving quickly through that same space.

Autonomy

The degree of autonomy has some of the most significant and varied long-term consequences for system design: the higher the overall system's autonomy, the less the human involvement and the greater the need for extensive and sophisticated processing inside the system. Such autonomy increases the need for multiple sensory modalities, translation between external requests and internal processing, and the internal computational model's complexity. Detailed characterization systems are relatively low-autonomy systems, because their intent is to simply deliver sensory information to a human user or external program. Event detection requires far more autonomy because the definition of interesting events must be programmed into the system, and the system must execute more complex queries and computations internally to process detailed measurements and identify events. Perhaps more than any other dimension, autonomy is most significant in moving us from embedding instruments to embedding computation in our physical world.

Modalities. Truly autonomous systems depend on multiple sensory modalities. Different modalities provide noise resilience to one another and can combine to eliminate noise and identify anomalous measurements.

Complexity. Greater system autonomy also

entails greater complexity in the computational model. A system that delivers data for human consumption leaves most of the computation to the human consumer or to a centralized program that can operate based on global information. A system that executes contingent on system state and inputs over time must execute a general programming language that refers to spatially and temporally variable events.

Where are we now?

Weiser often described the need for a range of different-sized devices (not just PCs and laptops, but devices the size of windows or scraps of paper). Surely he imagined them scaling down to the size of a pin or up to the size of an entire building. This section describes developments and trends that are key to realizing his vision of increasingly pervasive and invisible computing systems. In each of the following areas, we see a consistent trend from highly engineered deployments of modest scale using application-specific devices to ad hoc deployments of immense scale based on reusable components intended for system evolution. We will emphasize the role that these developments play in supporting system scale, variability, and autonomy.

Small packages in the physical world

A major theme in Weiser's work was the creation of small devices used in a larger intelligent environment. The Xerox ParcTab provided limited display, user input, and infrared communication in a palm-sized unit. Small, attachable RF identification (RFID) tags helped determine the approximate position and identity of the tagged object or individual in a space equipped with specialized readers.[7] Today, we see many consumer devices in the palm form factor possessing roughly the processing and storage capabilities of early-to-mid 1990s PCs. Personal digital assistants and pocket PCs have gained wireless connectivity either as variations of pager networks or 802.11 wireless LANs. However, both approaches have significant drawbacks—the former is expensive, has low bandwidth, and is void of proximity information, and the latter consumes too much energy and requires a large battery pack. There is now a strong push to incorporate Bluetooth short-range wireless networks into handheld devices, and numerous attractive radio technologies are on the horizon.

Recently these devices have gained a rich set of input modes besides user buttons and knobs. Most have acoustic input and output, and some incorporate accelerometers to detect gestures, orientation, or video input. Simultaneously, many cell phones have gained Internet browsing capability, and PDAs have gained cell phone capabilities for data and voice access. Soon all cell phones will know where they are thanks to GPS, which will bring a degree of location awareness into various personal computing devices.

In most existing ubiquitous computing environments, the sensing capabilities of these devices is used to detect human actions—to recognize gestures or the relationship between objects.[8,9] The network communicates the occurrence of these actions to the computers in the infrastructure that can process them. We can connect rich sensor arrays and sophisticated motor controllers to these devices as we would to a PC in a traditional process control environment (as on a manufacturing line) or to an embedded controller (as in an automobile). However, increased miniaturization raises the possibility of every tiny sensor and controller having its own processing and communication capabilities, such that the aggregate performs sophisticated functions. A sequence of University of California, Los Angeles wireless integrated networked sensor nodes demonstrates early examples of such wireless integrated sensors.[10] As these become small and numerous, we can place them close to the physical phenomenon of interest to provide tremendous detail and accuracy. This ability provides richness to ubiquitous environments through new modalities (for example, detecting user frustration by change in body temperature), but also enables the instrumentation of physical sites that humans can't access, such as the inside of structures, equipment, or aqueous solutions.

In addition to packing ever more computing and storage capacity onto a chip, decreasing lithographic feature size squeezes the same capacity into a progressively smaller area with an even greater reduction in power consumption. However, the ability to implement the radio or optical transceivers in the same module as the processor has provided a qualitative advance in size and power. Sensors and actuators have undergone a revolution with the emergence of microelectromechanical systems (MEMS) technology, in which mechanical devices such as accelerometers, barometers, and movable mirrors are constructed at minute size on a silicon chip using lithographic processes similar to those for integrated circuits. Improved equipment and technology result in smaller MEMS devices, lower power consumption, and improved device performance. Although these improvements are not as predictable or clearly tied to feature size as processing and storage are, a strong correlation exists. The node subsystems' size and performance improvements reduce power consumption and allow a corresponding decrease in the size and cost of the power supply as well. There have also been substantial improvements in battery technology, with improved storage density, form factor, and recharging, as well as the emergence of alternative storage devices, such as fuel cells and energy-harvesting mechanisms (see the "Energy" sidebar).

Several research groups are exploring how to build intelligent, information-rich physical environments by deploying many small, untethered, deeply integrated nodes. Core challenges involve designing systems to operate at low power consumption, storing and obtaining that power, orchestrating nodes to form larger networks, and deploying applications over fine-grain networks. At the far extreme, researchers have conducted design studies on the feasibility of building an entire system, including power storage, processing, sensing, and communication, in a cubic millimeter.[11–13] This scale should be achievable in practice five to 10 years out. Researchers have also made substantial progress in low-power CMOS radios at several performance points. The research community now

Energy

Energy constraints dominate algorithm and system design trade-offs for small devices. Energy storage has advanced substantially, but not at the pace we associate with silicon-based processing, storage, and sensing. Batteries remain the primary energy storage devices, although fuel-based alternatives with high energy density are being actively developed. As a general rule of thumb, batteries store approximately a joule per mm^3, but many factors influence the choice of technology for a particular application. Over the past 20 years, an AA nickel alkaline (NiCd and NiMH) battery's capacity has risen from 0.4 to 1.2 Amp hours with fast recharging (see http://books.nap.edu/books/0309059348/html/index.html). Lithium batteries offer higher energy density with fewer memory effects but longer recharge times. The zinc-based batteries used in hearing aids have high energy density but high leakage, so they are best for high usage over short duration. Recent polymer-based batteries have excellent energy density, can be manufactured in a range of form factors, and are flexible, but they are also expensive. Numerous investigations have focused on thin and thick film batteries, and researchers have fabricated tiny, 1-mm^3 lead-acid batteries, so we can expect to package energy storage directly with logic.

Fuel cells potentially have 10 times the energy density of batteries, considering just the fuel, but the additional volume of the membrane, storage, and housing lowers this by a factor of two to five. MEMS approaches are exploring micro heat engines and storing energy in rotating micromachinery. Solar panels remain the most common form of energy harvesting, but numerous investigations are exploring avenues for harvesting the mechanical energy associated with specific applications, such as flexing shoes, pushing buttons, window vibration, or airflow in ducts (see www.media.mit.edu/context, www.media.mit.edu/physics, or www.media.mit.edu/resenv). With existing technology, a cubic millimeter of battery space has enough energy to perform roughly 1 billion 32-bit computations, take 100 million sensor samples, or send and receive 10 million bits of data. As all the system's layers become optimized for energy consumed per operation, these numbers will increase by at least an order of magnitude and some by several orders.[1]

Sample battery energy ratings include

- Nonrechargeable lithium: 2,880 J/cm^3
- Zinc-air: 3,780 J/cm^3 (has very high leakage)
- Alkaline: 1,190 J/cm^3
- Rechargeable lithium: 1,080 J/cm^3
- Nickel metal hydride (NiMHd): 864 J/cm^3
- Fuel cells (based on methanol): 8,900 J/cm^3
- Hydrocarbon fuels (for use in micro heat engines): 10,500 J/cm^3

Sample scavenging energy ratings include

- Solar (outdoors midday): 15 mW/cm^2
- Solar (indoor office lighting): 10 uW/cm^2
- Vibrations (from microwave oven casing): 200 uW/cm^3
- Temperature gradient: 15 uW/cm^3 (from a 10° C temperature gradient)

REFERENCE

1. L. Doherty et al., "Energy and Performance Considerations for Smart Dust," *Int'l J. Parallel Distributed Systems and Networks*, vol. 4, no. 3, 2001, pp. 121–133.

widely uses a one-inch scale platform (see the "University Research" sidebar), which provides an opportunity to explore the node system architecture. It must be extremely energy efficient, especially in the low duty-cycle vigilance mode, and it must be extremely facile with event bursts. It must meet hard real-time constraints, such as sampling the radio signal within bit windows, while also handling asynchronous sensor events and supporting localized data processing algorithms. It also must be robust and reprogrammable in the field.

During the growth in capability and complexity of these devices, several distinct operating systems approaches have emerged to make application design more manageable. Real-time operating systems such as Vxworks (www.windriver.com), GeoWorks (www.geoworks.com), and Chorus (www.sun.com/chorusos) have scaled down their footprints and added TCP/IP capabilities, whereas Windows CE has sought to provide a subset of the familiar PC environment. PalmOS successfully focused on data element exchange with infrastructure machines, but provided little support for the concurrency associated with interactive communication. As the devices reach a processing and storage capability beyond the early workstations, compact Unix variants, especially Linux, have gained substantial popularity while providing real-time support in a multitasking environment with well-developed networking.

To make the networked embedded node an effective vehicle for developing algorithms and applications, a modular, structured runtime environment should provide the scheduling, device interface, networking, and resource management primitives on which the programming environments rest. It must support several concurrent flows of data from sensors to the network to controllers. Moreover, microsensor devices and low-power networks operate bit by bit (or in a few cases, byte by byte), so software must do much of the low-level processing of these flows and events. Often, operations must be performed within narrow jitter windows, such as when sampling the RF signal.

The traditional approach to controller design has been to hand-code scheduling loops to service the collection of concurrent flow events, but this yields brittle, single-use firmware that has poor adaptability. A more general-purpose solution is to provide fine-grain multithreading. Although

University Research

Research conducted through the University of California, Berkeley's DARPA project on SenseIT and ubiquitous computing produced a widely-used microsensor node. CrossBow (www.xbow.com) currently produces it in volume. The core building block is a 1-inch x 1.5-inch motherboard comprising a low-power microcontroller, low-power 900-MHz radio, nonvolatile memory, LEDs, network programming support, and vertical expansion bus connector. The microcontroller contains Flash program and SRAM data storage, analog digital converter, and external I/O (standard and direct ports). A second small microcontroller lets the node reprogram itself from network data. The sensors and actuators on the motherboard are associated with its own operation: battery voltage sensor, radio signal strength sensing and control, and LED display. The microcontroller's external interface is exposed in a standardized form on the expansion connector, providing analog, digital, direct I/O, and serial bus interconnections. Sensor packs for the specific applications, including termistors, photo detectors, accelerometers, magnetometers, humidity, pressure, or actuator connections, are stacked like tiny PC104 boards (see Figure A). The processor dissipates several nano-Joules per 8-bit instruction.

The sensor board for the Berkeley platform consists of five different microsensor modules to support several potential applications.

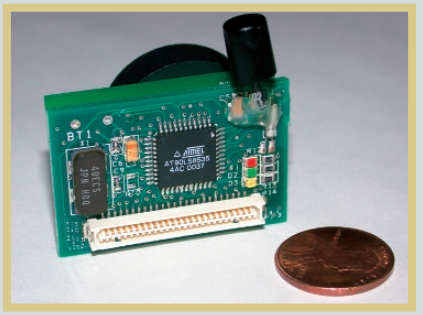


Figure A. The University of California, Berkeley, mote.

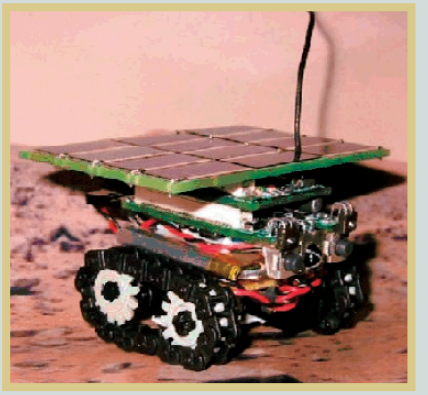


Figure B. The Robomote: A mobile sensor node.

The types of sensors it supports include light, temperature, acceleration, magnetic field, and acoustic, each of which is available off the shelf. All modules in the sensor board power cycle independently and are power isolated from the MICA's processor through an analog switch. Finally, the gain of the magnetometer and the microphone amplification is adjustable by tuning the two digital potentiometers over the I^2C bus.

We recently stacked a motor control board on a microsensor node and mounted the resulting assembly on a motorized chassis with two wheels.[1] The motor control board regulates wheel speeds and provides range information from two forward and one rear-looking infrared emitters. The resulting node (see Figure B) is essentially a small mobile robot platform with the same network interface as the microsensor nodes.

REFERENCE

1. G.T. Sibley et al., "Robomote: A Tiny Mobile Robot Platform for Large-Scale Sensor Networks," to be published in *Proc. IEEE Int'l Conf. Robotics and Automation*, 2002; http://robotics.usc.edu/projects/robomote.

researchers have studied this approach extensively for general-purpose computation, we can attack it even more effectively in the tiny, networked sensor regime, because the execution threads that must be interleaved are simple. These requirements have led to a component-based *tiny operating system* environment,[14] which provides a framework for dealing with extensive concurrency and fine-grain power management while providing substantial modularity for robustness and application-specific optimization. The TinyOS framework establishes the rules for constructing reusable components that can support extensive concurrency on limited processing resources.

Sensing and actuation

Interfacing to the physical world involves exchanging energy between embedded nodes and their environments. This takes two forms: sensing and actuation. Whatever the sensed quantity (temperature, light intensity), the sensor transducers a particular form of energy (heat, light) into information. Actuation lets a node convert information into action, but its main role is to enable better sensing. An actuator moves part of itself, relocates spatially, or moves other items in the environment. Sensing and actuation are together the means of physical interaction between the nodes and the world around them.

The 1990s saw MEMS technology transformed from a laboratory curiosity into a source of widespread commercial products. Millimeter-scale silicon accelerometers joined their cousins the silicon pressure sensors under the hoods of most automobiles,

and gyros and flow sensors are now also becoming common. Projection display systems with a million moving parts on a single chip are commonplace. Internet packets bounce off submillimeter mirrors that switch photons between different optical fibers. MEMS technologies are successful in the optical applications space because they provide far superior performance or the same performance at lower prices; they are successful in Detroit because they are reliable and dirt-cheap.

A common problem in both sensing and actuation is uncertainty. The physical world is a partially observable, dynamic system, and the sensors and actuators are physical devices with inherent accuracy and precision limitations. Thus sensor-measured data are necessarily approximations to actual values. In a large system of distributed nodes, this implies that we need some form of filtering at each node before we can meaningfully use the data. We can also achieve increased accuracy and fault tolerance by redundancy, using sensors with overlapping fields of view. This raises interesting challenges of sensor placement and fusion, especially in the context of very large networks. In addition to uncertainty, there is the further problem of latency in actuation. For closed loop control, stochastic latency can cause instability and unreliable behavior.

Although not traditionally associated with ubiquitous computing, robotics and MEMS developments play a critical role when it comes to sensing, actuation, and control. A particularly significant development in robotics within the past decade has been the move away from disembodied, traditional AI to real-time embedded decision making in physical environments. For nearly two decades, the dominant paradigm in mobile robotics research involved the offline design of control algorithms based on deliberation. These planner-based algorithms relied on logic and models of the robots and their environments, but the systems were unresponsive and slow to adapt to dynamic environments. Their dysfunctionality in physical domains spurred the development of control techniques that closely coupled perception and action with remarkable success. The earliest examples of these stateless, reactive systems were responsive but lacked generality and the ability to store representation. Modern, behavior-based control[15] generalizes reactive control by introducing the notion of behavior as an encapsulated, time-extended sequence of actions. Perception and action are still coupled tightly as in reactive systems, with the added benefits of representation and adaptation without any centralized control. An alternative modern approach is to hybridize control,[16] where a planner and a reactive system communicate through a third software layer designed explicitly for that purpose.

The physical world is a partially observable, dynamic system, and the sensors and actuators are physical devices with inherent accuracy and precision limitations.

Localization

For a system to operate on input from the physical world, nodes must know their location in three spaces to provide context-aware services to other system elements. A static or mobile node could answer the question "Where am I?" in several ways: the answer might be relative to a map, relative to other nodes, or in a global coordinate system. For a sensor network, this is particularly relevant, because we must often tag the queries to which it will provide answers based on sensor measurements with location information.

Localization is a main part of registration between the virtual and physical worlds. In a sensor network, this can take a variety of forms. Nodes in a network might report data tagged with relative location information, but from time to time, some nodes in the network might have to reference their data to an external anchor frame. Another example involves aggregation. Imagine two cameras with overlapping fields of view. A node aggregating data from these two cameras might perform the equivalent of stereo processing, but to do so, it must know the baseline between the two nodes—the location of one relative to the other.

Scale and autonomy play an important role in location computation. Several early large-scale systems relied on careful offline calibration and surveying before node deployment to ensure reliable localization. Examples include the Active Badge and Active Bat systems (see the "Sensing Location and Movement of People and Devices" sidebar). In robotics, however, the focus was on techniques for localizing small numbers of robots autonomously—without relying on presurveyed maps, beacons, or receivers.[17] A recent trend has been to investigate algorithms that successfully localize large-scale networks of nodes autonomously.

We can taxonomize localization techniques for embedded devices in several ways: the sensors used, whether the localization is with reference to a map, and whether the localization is done using external beacons and such.[18] Broadly speaking, we can view localization as a sensor-fusion problem. Given disparate sources of information about different aspects of node location and position, the problem is to design algorithms that can rationally combine the data from these sources to maintain an estimate of node location. In many interesting applications (with large numbers of small nodes), nodes cannot be placed with great care. In some systems, nodes might be mobile because they are robotic or because other entities in the environment can move them. Thus in-network, autonomous localization is a must for many real-world applications. This is a departure from most existing and planned ubiquitous computing systems for traditional office environments in which significant effort was expended upfront to instrument the environment for localization.

Several algorithms exist for localizing both fixed and mobile network nodes. One example recently demonstrated coarse localization of networks of mobile nodes by letting them build a map of their environment.[19] Significantly, the system localizes the nodes adaptively by updating an internal sparse representation of a changing environment. The algorithm uses the nodes themselves as landmarks, and could thereby allow the robot to determine each node's location without reference to environmental features or to a map.[20] The algorithm constructs a mesh in which nodes are represented by point masses, and springs represent observed relationships between nodes.[20] A relaxation algorithm determines the lowest energy state of the mesh and thereby the most probable location of the nodes.

A distributed system architecture

Looking forward, the distributed system architecture will make or break our ability to effectively instrument the physical world. Moreover, the same characteristics that enable these systems (miniaturization and wireless communication) impose constraints that require significant changes in the overall architecture to achieve the desired functionalities and properties.

First and foremost are the constraints imposed by having to operate within the limits on finite or slowly charging batteries. Remember that this energy's primary consumer in the context of low-power communication in physically complex settings is wireless communication.[10] Con-

Sensing Location and Movement of People and Devices

Many of the earliest ubiquitous computing systems focused on providing localization services using physically embedded, networked systems. These systems exhibited the characteristics of several nodes (scale), but otherwise differed significantly from current trends in that they had a relatively fixed—not ad hoc or variable—system structure. The application emphasis was on providing context and location to human applications with simple forms of autonomy relative to those anticipated in future systems.

Researchers took the first steps in the direction of interacting with the physical world as a general computing concept at the Xerox Parc and Olivetti (now AT&T) Research Lab in Cambridge. That work's emphasis was on building a range of different sized devices that could occupy distinct niches in the ecosystem of interactions with information. Much of the work focused on understanding how people might interact with such devices and how user interfaces could use contextual information about the collection of devices. Thus, the primary connection with the physical world involved determining each device's location and thereby its geometric relationship to others.

The Active Badge system, developed between 1989 and 1992, used an IR beacon that carried identity information from small mobile devices to IR sensors in the environment. Given a map of the physical space with sensors as landmarks, its short range and inability to pass through solid objects provided a convenient means of determining approximate location by proximity. A low-bandwidth wired network connected the sensors to a centralized processing capability, which could maintain a representation of all the tagged objects and their interrelationships to direct actions of various sorts. Typical examples were telephone connections and computing environments tracking the movement of individuals through a space. Passive RFID tags provide a similar capability in small, low-cost packages.[1]

More recently, the Active Bat system used trilateration of ultrasonic beacons to provide accurate position determination in a physical space.[2] Receivers are placed in a regular grid in the ceiling tiles. An RF beacon informs a particular mobile device to emit an ultrasonic beacon and start a time-of-flight measurement at the receivers. Arrival times of the pulse's edge are conveyed over a wired network from the receivers to a central PC, which can calculate the specific device's position. The largest system deployed uses 720 receivers to cover an area of 1,000 m^2 on three floors and can determine the positions of up to 75 objects each second to within a few centimeters. Thus, we see in the localization technology's advancement the use of a richer set of sensor modes and the integration of communication with the sensing and control process.

The Massachusetts Institute of Technology Media Lab and the Georgia Institute of Technology Aware Home projects have developed more direct means of sensing the interaction of people with their environment. These efforts include floor sensors to determine individuals' positions and movement (with the hope of determining identity from step patterns), weight and acoustic sensors in doorways, tag readers embedded in tables to determine the pattern of tagged objects, and numerous physical objects with various sensors, display capabilities, and tags. A representation of all this information is projected into a higher tier, which deals with proxies of the physical objects as widgets. Hewlett Packard's Cooltown seeks to provide an infrastructure for building applications using many such instrumented devices interfacing to various PDAs and mobile computers.

REFERENCES

1. R. Want et al., "Bridging Physical and Virtual Worlds with Electronic Tags," *Proc. Conf. Computer-Human Interaction*, ACM Press, New York, 1999.

2. A. Harter et al., "The Anatomy of a Context-Aware Application," *Proc. 5th Ann. ACM/IEEE Int'l Conf. Mobile Computing and Networking* (MobiCom), ACM Press, New York, 1999, pp. 59–68.

Directed Diffusion

Directed Diffusion was designed to promote adaptive, in-network processing in wireless sensor networks. In Directed Diffusion, sensors publish and clients subscribe to data. Both identify data by attributes such as "the southwest" or "acoustic sensors." Diffusion divorces data identity from node identity. To do this, it uses a simple typing mechanism and encoding of attribute-based naming with simple matching rules. Names are sets of attributes, each a tuple, including keys, values, and operations.

As an example, the query, "sensor EQ seismic, latitude GT 100, latitude LT 101" would trigger a sensor with data tagged, "sensor IS seismic, latitude IS 100.5." Matching is not meant to be a general-purpose language. User-provided code or filters can be distributed to the sensor network to perform application-specific, in-network processing tasks such as data aggregation, caching, and collaborative signal processing.[1]

Directed Diffusion was designed and implemented by researchers at the University of Southern California's Information Sciences Institute and is now used by multiple research projects to support collaborative signal processing applications as part of the DARPA SenseIT program. Directed Diffusion is supported under Linux and runs on off-the-shelf embedded PC devices. A constrained subset of Diffusion runs under TinyOS on the University of California, Berkeley's motes described earlier in this article.These implementations support a common API, which has been used by Cornell, BAE Systems, Xerox PARC, and Pennsylvania State University collaborative processing applications run in experimental field trials. Diffusion is also supported in the widely used ns-2 network simulator and supports APIs identical to the Linux implementations.

REFERENCE

1. J. Heidemann et al., "Building Efficient Wireless Sensor Networks with Low-Level Naming," *ACM Symp. Operating Systems Principles*, ACM Press, New York, 2001.

sequently, we cannot realize long-lived autonomous systems by simply streaming all the sensory data out of the nodes for processing by traditional computing elements. Rather, computation must reside alongside the sensors, so that we can process time-series data locally. Instead of building a network of sensors that all output high-bandwidth bit streams, we must construct distributed systems whose outputs are at a higher semantic level—compact detection, identification, tracking, pattern matching, and so forth.

To support such an architecture, important systems-oriented trends are emerging. Two important examples are self-configuring networks and data-centric systems. One objective of self-configuration is to let systems exploit node redundancy (density) to achieve longer unattended lifetimes. Techniques for coordinating and adapting node sleep schedules to support a range of trade-offs between fidelity, latency, and efficiency are also emerging.[21–24] Ad hoc routing techniques represent an even earlier example of self-configuration in the presence of wireless links and node mobility,[25] but they still support the traditional IP model of shipping data from one edge of the network to another. Small form factor wireless sensor nodes cannot afford to ship all data to the edges for outside processing, and fortunately do not need to operate according to the same layering restrictions as IP networks when it comes to application-layer data processing at intermediate hops. Directed diffusion promotes in-network processing by building on a data-centric instead of address-centric architecture for the distributed system or network. Using data naming as the lowest level of system organization supports flexible and efficient in-network processing (see the "Directed Diffusion" sidebar).[26]

A second trend is the increased reliance on tiered architectures in which fewer higher-end elements complement the more limited capabilities of widely and densely dispersed nodes. Very small devices will inevitably possess limited storage and computing resources as well as limited bandwidth with which to interact with the outside world. Introducing tiered architectures, where some system elements have greater capacity, is desirable. In a tiered architecture, the smallest system elements help achieve spatial diversity and short-range sensing, whereas the computationally powerful elements implement more sophisticated and performance intensive processing functions, such as digital signal processing, localization, and long-term storage. These higher-tier resources could include robotic elements that traverse the sensor field, delivering energy to depleted batteries, or that compute localization coordinates for ad hoc collections of smaller nodes.

Where are we headed?

As we embark on Weiser's vision, we are discovering that the ability to form networks of devices interacting with the physical world opens broad avenues for information technology beyond the highly connected responsive home or workspace. We foresee thousands of devices embedded in the civil infrastructure (buildings, bridges, water ways, highways, and protected regions) to monitor structural health and detect crucial events. Eventually, such devices might be tiny enough to pass through bodily systems or be usable in large enough numbers to instrument major air or water flows. In the nearer term, embedded sensor networks can fundamentally change the practice of numerous scientific endeavors, such as studies of complex ecosystems, by providing in situ monitoring and measurement at unprecedented levels of temporal and spatial density without disturbing the complex systems under study.[27]

The tremendous progress toward miniaturization means that we can put instruments into the experiment, rather than conducting the experiment within an instrument. For the laboratory, this means wireless sensors for measurement and log-

ging every test tube and beaker. For the ubiquitous computing environment, it means sprinkling sensing capabilities through a space appropriate to the activities of interest, rather than conducting studies in a specially designed testbed environment. Realizing this dimension of Weiser's long-term vision will require more than dramatic developments in hardware miniaturization. It will require system-level advances including the programming model, closed loop control, predictability, and environmental-compatibility.

We first need a system-wide architecture that supports interrogating, programming, and manipulating the physical world. Some of these systems will require extensive in-network compression and collaborative processing to exploit the spatial and temporal density of emerging systems. One emerging characteristic of such an architecture is the shift to naming data in terms of the relevant properties of the physical system instrumented—naming data instead of nodes.[26,28] Many systems will be organized around spatial and temporal coordinates. Given a tuple-space for the instrumented environment, we'll need a programming model for the computations distributed in time and space. Various models are under consideration and alternatively view the system as a distributed database or a loosely coupled parallel computing structure.[29]

Increasingly, physically embedded systems will need to self-organize. Self-organization, particularly *spatial* reconfiguration, is needed to address variability at multiple scales. An interesting (and unique) aspect of the interaction with the physical world is the ability to manipulate it. Although today's systems are built with self-configuration in mind, in the longer term, these systems must integrate manipulation and reconfigure themselves and the world in which they are embedded. Closed-loop control will be contained in the distributed system and is a formidable challenge due to the inherent stochastic communication delays in such systems.

As the systems become increasingly autonomous and grow to include actuation, the need for predictability and diagnosability will be a critical stumbling block. Weiser's vision requires these systems ultimately to disappear into the environment, but they must do so while operating correctly. Users will not rely on them if they do not degrade with predictable behaviors or if they do not lend themselves to intuitive diagnosis, which comes with the ability to develop mental models.[6]

There is the potential for damage or intrusion on the spaces instrumented. Two examples whose risks are evident today are the violation of personal privacy through lack of anonymity in instrumented spaces and the generation of "space garbage" by leaving behind depleted nodes in the environment. Techniques for anonymity-preserving systems and for harvesting energy from the environment are enabling technologies, but such issues ultimately require social and legal policy.

As the systems become increasingly autonomous and grow to include actuation, the need for predictability and diagnosability will be a critical stumbling block.

Moving into a ubiquitous computing world calls for the computing community to embrace interdisciplinary approaches and to transform the educational process to enable deep interdisciplinary advances. Interfacing to the physical world is arguably the single most important challenge in computer science today. The frontier of almost any traditional CS subdiscipline is in this area. Examples of the disciplines involved include networking, operating system, databases, artificial intelligence, virtual reality, and signal processing.[3] An implication of this area's interdisciplinary nature is the need for upgrading training of our students at both the undergraduate and graduate levels. Students need skills that extend beyond constructing complex programs to manipulate virtual objects. They need the ability to understand and model physical world processes that they will have to measure and with which they will have to cope.[6] P

ACKNOWLEDGMENTS

David Culler's work is supported in part by the Defense Research Projects Agency (NEST F33615-01C-1895), National Science Foundation (RI EIA-9802069), California MICRO, Compaq, and Intel. Deborah Estrin's research is supported in part by grants from the Defense Research Projects Agency (GALORE F33615-01-C-1906 and SenseIT DABT 63-99-1-0011), NSF (SCOWR EIA-0082498), and Intel. Kris Pister's research is supported by grants from DARPA MTO and ITO. Gaurav Sukhatme's research is supported in part by DARPA (MARS DABT63-99-1-0015), the National Science Foundation (SCOWR ANI-0082498 and ITR EIA-0121141), NASA (1231521), the Department of Energy (RIM DE-FG03-01ER45905), and Intel.

REFERENCES

1. D. Estrin et al., "Next Century Challenges: Scalable Coordination in Sensor Networks," *Proc. ACM Conf. Mobile and Computing Networking* (MobiCom), ACM Press, New York, 1999.

2. M. Chu, H. Haussecker, and F. Zhao, "Scalable Information-Driven Sensor Querying and Routing for Ad Hoc Heterogeneous Sensor Networks," to be published in *Int'l J. High Performance Computing Applications*, 2002.

3. F. Zhao, J. Shin, and J. Reich, "Information-Driven Dynamic Sensor Collaboration for Target Tracking," to be published in *IEEE Signal Processing*, 2002; www.parc.xerox.com/spl/projects/cosense/pub/ieee_spm.pdf.

4. S. Pradhan, J. Kusuma, and K. Ramchandran, "Distributed Compression in a Dense Sensor Network," to be published in *IEEE Signal Processing*, 2002.

5. D. Li et al., "Detection, Classification and Tracking of Targets in Distributed Sensor Networks," to be published in *IEEE Signal Processing*, 2002.

6. Computer Science and Telecommunication

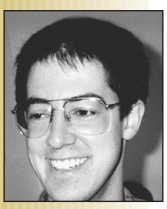
Board, *Embedded Everywhere: A Research Agenda for Networked Systems of Embedded Computers*, National Research Council, Washington, D.C., 2001.

7. R. Want et al., "The Active Badge Location System," *ACM Trans. Information Systems*, vol. 10, no. 1, Jan. 1992, pp. 91–102.
8. C.D. Kidd et al., "The Aware Home: A Living Laboratory for Ubiquitous Computing Research," *Proc. 2nd Int'l Workshop Cooperative Buildings*, Lecture Notes in Computer Science, vol. 1670, Springer-Verlag, Berlin, 1999, pp. 190–197.
9. T. Kindberg and J. Barton, "A Web-Based Nomadic Computing System," *Computer Networks*, vol. 35, no. 4, Mar. 2001, pp. 443–456.
10. G. Pottie and W. Kaiser, "Wireless Integrated Network Sensors," *Comm. ACM*, vol. 43, no. 5, May 2000, pp. 51–58.
11. J.M. Kahn, R.H. Katz, and K.S.J. Pister, "Emerging Challenges: Mobile Networking for Smart Dust," *J. Comm. and Networks*, vol. 2, no. 3, Sept. 2000, pp. 188–196; www.eecs.berkeley.edu/~jmk/pubs/jcn.00.pdf.
12. K.S.J Pister and B.E. Boser, *Smart Dust: Wireless Networks of Millimeter-Scale Sensor Nodes*, Electronics Research Laboratory Research Summary, Electronic Research Lab, Univ. of California, Berkeley, 1999.
13. B. Warneke et al., "Smart Dust: Communicating with a Cubic-Millimeter Computer," *Computer*, vol. 34, no. 1, Jan. 2001, pp. 44–51.
14. J. Hill et al., "System Architecture Directions for Networked Sensors," *ACM Architectural Support for Programming Languages and Operating Systems*, ACM Press, New York, 2000.
15. M.J. Mataric, "Behavior-based Control: Examples from Navigation, Learning, and Group Behavior," *J. Experimental and Theoretical Artificial Intelligence*, vol. 9, nos. 2–3, 1997, pp. 323–336; www-robotics.usc.edu/~maja/publications/jetai-arch.ps.gz.
16. R. Arkin, "Toward the Unification of Navigational Planning and Reactive Control," *Proc. AAAI Spring Symp.*, AAAI, Menlo Park, Calif., 1989.
17. S. Thrun et al., "Robust Monte Carlo Localization for Mobile Robots," *Artificial Intelligence*, vol. 101, 2000, pp. 99–141; www-2.cs.cmu.edu~thrun/papers/thrun.robust-mcl.html.
18. J. Hightower and G. Borriello, "Location Systems for Ubiquitous Computing," *Computer*, vol. 34, no. 8, Aug. 2001, pp. 57–66.
19. G. Dedeoglu and G.S. Sukhatme, "Landmark-based Matching Algorithm for Cooperative Mapping by Autonomous Robots," *Proc. 5th Int'l Symp. Distributed Autonomous Robotic Systems*, Springer-Verlag, Berlin, 2000, pp. 251–260; www-robotics.usc.edu/~gaurav/Papers/goksel-dars.ps.gz.
20. A. Howard, M.J. Mataric, and G.S. Sukhatme, "Relaxation on a Mesh: A Formalism for Generalized Localization," *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems* (IROS), IEEE CS Press, Los Alamitos, Calif., 2001, pp. 1055–1060.
21. B. Chen et al., "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks," *Proc. 7th ACM Conf. Mobile and Computing Networking* (MobiCom), ACM Press, New York, 2001.
22. A. Cerpa and D. Estrin, "ASCENT: Adaptive Self-Configuring Sensor Network Topologies," to be published in *Proc. IEEE Infocom*, IEEE Press, Piscataway, N.J., 2002.
23. C. Schurgers, V. Tsiatsis, and M. Srivastava, "STEM Topology Management for Efficient Sensor Networks," to be published in *Proc. IEEE Aerospace Conf.*, IEEE CS Press, Los Alamitos, Calif., 2002.
24. Y. Xu, J. Heidemann, and D. Estrin, "Geography-Informed Energy Conservation for Ad Hoc Routing," *Proc. 7th Ann. ACM/IEEE Int'l Conf. Mobile Computing and Networking* (MobiCom), ACM Press, New York, 2001.
25. D. Johnson and D. Maltz, "Protocols for Adaptive Wireless and Mobile Networking," *IEEE Personal Comm.*, vol. 3, no. 1, Feb. 1996.
26. J. Heidemann et al., "Building Efficient Wireless Sensor Networks with Low-Level Naming," *ACM Symp. Operating Systems Principles*, ACM Press, New York, 2001.
27. A. Cerpa et al., "Habitat Monitoring: Application Driver for Wireless Communications Technology," *Proc. ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean*, ACM Press, New York, 2001.
28. W. Adjie-Winoto et al., "The Design and Implementation of an Intentional Naming System," *ACM Symp. Operating System Principles*, ACM Press, New York, 1999.
29. P. Bonnet, J.E. Gehrke, and P. Seshadri, "Querying the Physical World," *IEEE Personal Comm.*, vol. 7, no. 5, Oct. 2000, pp. 10–15.

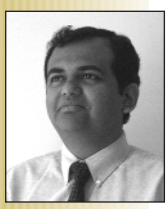
the AUTHORS

Deborah Estrin is a professor of computer science at the University of California, Los Angeles. Her research interests include design of network and routing protocols for large global networks, sensor networks, and applications for environmental monitoring. She has a PhD in computer science from the Massachusetts Institute of Technology. She has served on several program committees and editorial boards. Contact her at the Computer Science Dept., UCLA, 3713 Boelter Hall, 420 Westwood Plaza, Los Angeles, CA 90095; destrin@cs.ucla.edu.

David Culler is a professor of computer science at the University of California, Berkeley, and director of Intel Research at UCB. His research interests include embedded networks of small wireless devices, parallel computer architecture, parallel programming languages, and high-performance communication. He has a PhD from MIT. Contact him at Computer Science Division #1776, 627 Soda Hall, UCB, Berkeley, CA 94720-1776; culler@cs.berkeley.edu.



Kris Pister is an associate professor in the Electrical Engineering and Computer Sciences Department at UCB. His research interests include the development and use of standard MEMS fabrication technologies, micro robotics, and CAD for MEMS. He has a BA in applied physics from the University of California, San Diego, and an MS and PhD in electrical engineering from UCB. Contact him at the Electrical Eng. and Computer Science Dept., UCB, 497 Cory Hall, Berkeley, CA 94720-1770; pister@eecs.berkeley.edu.



Gaurav Sukhatme is an assistant professor in the Computer Science Department at the University of Southern California and the associate director of its Robotics Research Laboratory. His research interests include embedded systems, mobile robot coordination, sensor fusion for robot fault tolerance, and human–robot interfaces. He has an MS and PhD in computer science from USC. He is a member of the IEEE, AAAI, and ACM and has served on several conference program committees. Contact him at the Computer Science Dept., MC0781, USC, 941 West 37th Place, Los Angeles, CA 90089-0781; gaurav@usc.edu.