Instrumenting the physical world with pervasive networks

Deborah Estrin (UCLA)
David Culler (UCB)
Kris Pister (UCB)
Gaurav Sukhatme (USC)

Abstract

This paper addresses the challenges and opportunities of instrumenting the physical world with pervasive networks of sensor-rich, embedded computation. Such systems will 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, without explicit human input. Of particular importance to the technical community is to make such pervasive computing itself pervasive. We need reusable building blocks that allow us to move away from specialized instrumentation of each particular environment and toward building reusable techniques for sensing, computing, and manipulating the physical world. We present taxonomy of emerging systems and challenges and outline the enabling technological developments.

I. Introduction and Motivation

Mark Weiser envisioned a world in which computing is so pervasive that everyday devices are able to sense their relationship to us and to each other. They can, thereby, respond so appropriately to our actions that the computing aspects fade into the background. Underlying this vision, is the assumption that sensing of 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 Scientific American article; Sal looked out her window and saw "tracks" as evidence of her neighbors' morning strolls. What sort of system was implied by this seemingly simple functionality? Certainly Weiser was not envisioning ubiquitous cameras placed throughout the neighborhood, providing input to sophisticated vision processors. Such a solution would be far too heavy weight for the relatively casual nature of the application, as well as being quite invasive with respect to personal privacy. Instead, Weiser was positing the existence of far less intrusive instrumentation of the neighborhood spaces; perhaps smart paving stones that would be able to detect and indicate local activity, and indicate direction of the walker based on detections exchanged among neighboring nodes. As we have marched technology forward, we are now in a position to translate this aspect of the Weiser's vision to reality and apply it to a wide range of important applications, from both a computing and social perspective.

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 will 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, without explicit human input. Of particular importance to the technical community is to make such pervasive computing itself pervasive. We need reusable building blocks that allow us to move away from specialized instrumentation of each particular environment and 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 physical world meant deploying a highly-engineered collection of instruments placed to obtain particular inputs, and reporting the data over specialized wired control protocols to data acquisition computers (e.g., in support of a particular manufacturing or process control application). Even ubiquitous computing test-beds have retained much of this engineered data acquisition style, although they are used 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 a variety of application types.[Estrin99]

The underpinnings of this emerging frontier are technological: first, the processes and fabrication techniques that have driven the ‘Moore's Law’ growth of ever more powerful chips also allow a fixed level of computation or storage to move into ever more minute devices, second MEMS techniques allow a wide spectrum of sensors to be implemented cheaply in silicon, and finally low-power CMOS radio circuits and other forms of communication allow general purpose networks to be integrated directly into low-power devices. Computing, sensing, communicating and interacting with the physical world are converging in low cost, mainstream technology. During this decade, we can realistically expect to be designing systems comprising huge numbers of microscopic devices that interact with the physical world and with cyberspace.

As we embark on this venture 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, protected regions, etc.) to monitor structure health and detect crucial events. Eventually, such devices may be tiny enough to pass through bodily systems or useable 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 [Cerpa01].

The most serious impediments to such advances are **systems challenges**. The immense scale in terms of numbers of distributed system elements, the limited physical access, and the extreme environmental dynamics of this regime, when considered together, imply that familiar layers of abstraction, the kinds of hardware acceleration employed, even our algorithmic techniques, must be fundamentally reexamined.

- **Immense Scale**: These systems will be comprised of vast numbers of small physical size devices, To achieve dense instrumentation of complex physical systems, individual devices must scale down to extremely small volume, with applications formulated in terms of vast numbers of devices. In 5-10 years, complete systems with computing, storage, communication, sensing, and energy storage may be as small a cubic millimeter, but the capacity of each aspect will be limited. Fidelity and availability will come from the quantity of partially redundant measurements and their correlation, rather than quality and precision of the individual components

- **Limited Access:** Many devices will be embedded in the environment in places that are inaccessible or very expensive to connect with wires. Therefore, the individual system elements will be largely un-tethered, unattended, and resource constrained. Not only will much communication be wireless, nodes will also need to rely upon on board and harvested energy (e.g., batteries and solar cells). Inaccessibility, as well as sheer scale, implies that they must operate without human attendance. Thus, programming, management, and orchestration must be performed *en masse* and *in situ*. Each piece is such a small part of the whole, and no one can reasonably lay hands on all of them. At sufficient levels of efficiency, energy harvested from the environment can potentially allow arbitrary lifetimes, however, 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 need to be outdoors, if batteries cannot be recharged, they will seriously impact 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, these devices are capable of sensing their environment to provide inputs to higher-level tasks; which they may also work to perform. In addition, device performance is directly influenced by environmental changes. In particular, propagation characteristics of low power RF are dramatically influenced by environmental factors. These time-varying factors can effectively create mobility even in stationary configurations. These devices also experience *extreme variation* in demand: The vast majority of the time, they are observing 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. Redundant flows must be managed effectively. Consequently, extremely 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 the availability of resources and stimuli

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

In the remainder of this paper we present taxonomy of system types, which we expect to emerge during this next decade of research and development, summarize technological developments that are most relevant to the full range of systems, and conclude with some major research challenges.

II. Taxonomy of systems

The applications of physically-embedded networks are as varied as are the physical environments in which we live and work. And yet even with this heterogeneity, there are many opportunities for exploiting commonality across them. There are also many reasons to do so. Most important is the hope to achieve pervasiveness by enabling system reuse and evolution. Moreover, within any one environment or system, more than one “type” of system/application might be active; either as the system evolves, or from the outset. One step toward identifying common building blocks is to define taxonomy of systems and applications so that reusable and paramaterizable features can be identified and fostered.

We divide up the space of physically embedded systems along the critical dimensions of **space and time**. Within these dimensions we are interested in the **scale** (sampling, extent, and density), **variability** (including mobility), and **autonomy** (including complexity of internal computation). We address these dimensions with respect to both the environmental stimuli being captured, as well as the system elements themselves.

Spatial and temporal scale concerns both the sampling interval, and 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 are innovative collaborative signal processing techniques such as those described in [Chu02, Zhao02, Pradhan02, Li02]]. On the other hand, systems intended to operate over extended periods of time and extended 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, which are not encountered in low-density systems.

- **Spatial and temporal sampling** scale is ultimately dictated by the physical phenomena measured. 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 both of the phenomena and of the application. If the system is needed for event detection, the requirement may be more lax than if the goal is event/signal reconstruction. For example, if a structure is being monitored to detect structural faults, signatures of seismic response may be compared to “healthy” signatures at a relatively coarse grain; whereas if data is being collected in order to generate profiles of structure response, fine grain data is needed.
- **Spatial and temporal extent** of systems also varies widely. At the high end of this continuum are environmental monitoring systems in outdoor environments that may 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. However, elements of pervasive computing systems also extend to the smaller end of the continuum such as reconfigurable fabric that may be worn by users or deployed to monitor surfaces of structures and machinery.
- **System density** is a measure of sensor nodes per footprint of the 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 therefore opportunities to combine measurements to eliminate channel or coupling noise. Similarly, where density is high enough to allow over-sampling, nodes can be put to sleep for long periods of time and thereby extend coverage over time.

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 the system elements and/or the phenomena being sensed. Relatively static systems emphasize design time optimization whereas more variable systems must employ run time self-organization and may be fundamentally limited in the extent to which they can be both variable and long-lived.

- Ad hoc vs. engineered **system structure** refers to the variability in system composition [NRC-EE01]. At one extreme of this continuum is a system used to monitor a structure such as a building or bridge or even an individual airplane. At the other end are sensor networks deployed in remote regions to study bio-complexity. More traditional pervasive computing systems will embody aspects of both systems; elements of the systems such as instrumentation in an aware room or house may be relatively static; while the larger system that includes humans and subsystems carried in and out of the space is clearly ad hoc.
- Variability in **system task** determines the extent to which the system can be optimized for a single mode of operation. Even a structurally static system maybe be needed for different tasks over time; such as a structural system that periodically may be used to generate a structure profile, may act as an event monitoring system for long periods of time. Similarly a system used regularly to measure the effectiveness of a building-wide air conditioning system, may occasionally be required to integrate the inputs from new sensors used to detect traces of newly characterized toxins
- **Variability in space**, i.e., mobility, applies to both system nodes and phenomena. In many systems of interest, most or all of the nodes remain fixed in space once placed. However many interesting if longer term systems will include elements that move themselves or that are tied to objects that move them (e.g. vehicles or people). Similarly, the phenomena monitored by these systems 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 very different challenges than one designed to track humans moving quickly through that same space.

Finally, the degree of **autonomy** has some of the most significant and varied long-term consequences for system design: the higher the autonomy of the overall system, the less the human involvement, and the greater the need for extensive and sophisticated processing inside of the system itself. Such autonomy increases the need for multiple sensory modalities, translation between external commands/queries and internal processing, and the complexity of the internal computational model. Detailed characterization systems can be considered as relatively low autonomy systems because their intent is to simply deliver all or most sensory information to a human user or program outside of the system. 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 of the other dimensions, autonomy is the most significant in moving us from embedding instruments to that of embedding computation in our physical world.

- Truly autonomous systems will depend upon **multiple sensory modalities**. Different modalities provide noise resilience to one another and can often be combined to eliminate noise and identify anomalous measurements.

- Greater system autonomy will also entail greater **complexity in the computational model**. A system that delivers data for human consumption leaves most of the computation to the human consumer, or perhaps to a centralized program that can operate then 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.

There are common trends that are enabling the full range of system types. One such trend is the availability of large numbers of programmable, communication-enabled, devices. A second is the miniaturization of these devices, which raises both radical opportunities and challenges. Perhaps of greatest importance in the future is the development of system wide techniques for localization of system elements in three-space, and of system wide architectures designed to maximize overall system capability and lifetime by enabling increasingly sophisticated, adaptive, autonomous behaviors. Section III elaborates on these trends.

III. Where are we: Developments and Trends

Marc Weiser often described the need for a range of different sized devices; not just PCs and Laptops, but devices the size of scraps of paper or of windows. Surely he imagined them scaling down to the size of a pin or thread, or up to the entire building. This section describes developments and trends that are key to realizing the vision of increasingly pervasive and invisible ubiquitous computing systems: small autonomous devices that can bridge the physical world with communication networks, sensors and actuators which transform between information and energy, localization of where observed events take place, and data-centric distributed system organization to get information from where it is generated to where it is processed. In each of these 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 and intended for system evolution over time. We emphasize the role that these developments play in supporting system scale, variability and autonomy.

A. Small form factor, integrated packages that can be interfaced to the physical world

A major theme in the Weiser's work was the creation of small devices used in a larger intelligent environment. The Xerox PARC TAB provided limited display, user input, and IR communication in a palm-sized unit. Small, attachable, RFID tags were used to determine the approximate position and identity of the tagged object or individual in a space equipped with specialized readers[Want92]. Today we see a vast number of consumer devices in the palm form factor possessing roughly the processing and storage capabilities of early to mid-90s PCs. PDAs and PocketPCs have gained wireless connectivity either as variations of pager networks [Mercury]or 802.11 wireless LAN. However, both approaches have significant drawbacks, as the former is expensive, low bandwidth, and void of proximity information, whereas the latter consumes too much energy and requires a battery pack as large as that used for the rest of the device. 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, while some incorporate accelerometers to detect gestures and orientation or video input [PocketPCCamera]At the same time, many cellular phones have gained internet browsing capability, while PDAs gain cell phone capabilities for data a voice access. Soon all cell phones will know where they are via GPS, which will bring a degree of location awareness into a vast array of personal computing devices.

In most existing ubiquitous computing environments, the sensing capabilities of these devices is used to detect human actions, e.g., to recognize gestures, or the relationship between objects [Kidd99, Kindeberg01] and the network is used to communicate the occurrence of these actions to computers in the infrastructure that can process them in terms of the large application context. Rich sensor arrays and sophisticated motor controllers can be connected to these devices as they would to a PC in a traditional process control environment (as 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 sophisticated functions are performed by the aggregate. Early examples of such wireless integrated sensors were demonstrated by the sequence of UCLA WINS nodes [Pottie-Kaiser00]. As these become very small and numerous, they can be placed close to the physical phenomenon of interest to provide tremendous detail and accuracy. This ability provides richness to ubiquitous environments though new modalities (the user is getting frustrated, as indicated by change in temperature, skin moisture, heartbeat, or hand steady), but also makes possible the instrumentation of important physical sites that are inaccessible to humans, such as the inside of structures, equipment, or aqueous solutions; or on an extreme scale, such as individual plants on a large reserve or farm.

A smart autonomous node interfaced to the physical world comprises processing, data storage, communication hardware, a sensor and/or actuator, and a power supply. The size of each of these benefit from the improvements in the manufacturing of integrated circuits associated with Moore’s Law. Besides packing ever more computing and storage capacity into a chip, decreasing lithographic feature size allows the same capacity to be squeezed into a progressively smaller area with an even greater reduction in power consumption. Communication circuitry enjoys these advances as well, but in addition the ability to implement the radio or optical transceivers in the same module as the processor has provides a qualitative advance in size and power. Sensors and actuators have undergone a revolution with the emergence of micro electromechanical systems (MEMS) technology, in which mechanical devices like 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. While these improvements are not as predictable or as clearly tied to feature size as are processing and storage, there is a strong correlation nonetheless. The improvements in the size and performance of the node subsystems 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 Box 1).

Several research groups are exploring how to build intelligent, information rich physical environments by deploying many small, un-tethered nodes that are deeply integrated with the environment. Core challenges involve designing systems to operate at very low power consumption, storing and obtaining that power, orchestrating nodes to form larger networks, and deploying applications over such a fine-grain network. At the far extreme, design studies have been conducted on the feasibility of building an entire system, including power storage, processing, sensing, and communication in a cubic millimeter [Kahn00, Pister99, Warneke01]. It is anticipated that this scale is achievable in practice 5-10 years out. Substantial progress has been made in low-power CMOS radios at a variety of performance points. A one-inch scale platform is now in wide use within the research community (see Box 2). This provides an opportunity to explore the node system architecture upon which the networking capabilities and application logic of future pervasive environments may be carried out. 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 handling asynchronous sensor events and supporting localized data processing algorithms. It must be robust and re-programmable in the field.

With 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 (http://www.windriver.com), GeoWorks (http://www.geoworks.com), and Chorus (http://www.sun.com/chorusos/), that were developed for controlling instruments attached to PCs in embedded applications, have scaled down their footprint and added TCP/IP capabilities; while WindowsCE sought to provide a subset of the familiar PC environment. PalmOS was successful in focusing on data element exchange with infrastructure machines, but provides little support for the concurrency associated with interactive communication. As the devices have reached 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. For further discussion of hardware developments see Want et. al. in this issue.

To make the networked embedded node an effective vehicle for developing algorithms and applications, a modular, structured runtime environment provides the scheduling, device interface, networking, and resource management primitives upon which the programming environments rest. It must support several concurrent flows of data stream from sensors out to the network and to controllers. Moreover, micro-sensor devices and low-power networks operate bit-by-bit, or in a few cases byte-by-byte, so much of the low-level processing of these flows and events must be performed in software. Often, operations must be performed within narrow jitter windows, e.g., 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. While this approach has been studied extensively for general-purpose computation, it can be attacked even more effectively in the tiny-networked sensor regime, because the execution threads that must be interleaved are very simple. These requirements have led to a component-based “tiny OS” environment[Hill00], which provides a framework for dealing with extensive concurrency and fine-grained power management while providing substantial modularity for robustness and application specific optimization. Applications are described by a graph of components that are interconnected through narrow command and event interfaces The TinyOS framework establishes the rules for constructing components that can be reused and can support extensive concurrency on limited processing resources.

Begin Box 1

Energy constraints dominate algorithm and system design trade-offs for such small devices. Energy storage has advanced substantially, although 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 about a joule per mm^3, however, there are many factors that influence the choice of technology for a particular application. Over the past 20 years the capacity of a AA nickel alkaline (NiCd and NiMH) battery has risen from 0.4 to 1.2 Amp hours with fast recharging [Nap97]. Lithium batteries offer higher energy density with lesser memory effects, but longer recharge times. The zinc-based batteries used in hearing aids have high energy density, but high leakage, so they are suited to 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 are expensive. Numerous investigations have focused on thin and thick film batteries that can be deposited directly on the semiconductor die and tiny, one mm^{3,} lead-acid batteries have been fabricated, 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 currently lowers this by a factor of two to five. MEMS approaches are exploring micro heat engines and storing energy in rotating micro-machinery. Given that storage per unit area. Solar panels remain the most common form of energy harvesting, but numerous investigations are exploring avenues for harvesting mechanical energy associated with specific applications, such as flexing shoes, pushing buttons, vibration of windows, or air flow in ducts [MIT_media_lab]. With existing technology, a cubic millimeter of battery has enough energy to perform roughly a billion 32-bit computations, take 100 million sensor samples, or send and receiver 10 million bits of data. As all the layers of the system become optimized for energy consumed per operation, all of these numbers can increase by at least an order of magnitude, and some by several orders [Doherty01].

Sample Battery Energy Ratings:
Non-rechargeable lithium 2880 J/cm^3
zinc-air 3780 J/cm^3, but have very high leakage
Alkaline 1190 J/cm^3
rechargeable lithium 1080 J/cm^3
Nickel Metal Hydride (NiMHd) 864 J/cm^3
Fuel Cells (based on methanol) 8900 J/cm^3
Hydrocarbon fuels (for use in micro heat engines) 10500 J/cm^3

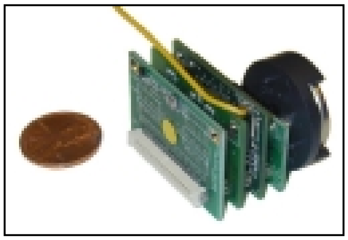
Sample Scavenging Energy Ratings:
Solar (outdoors midday) 15 mW/cm^2
Solar (indoor office lighting) 10 uW/cm^2
Vibrations (based on vibrations from microwave oven casing) 200 uW/cm^3
Temperature gradient 15 uW/cm^3 from a 10 degree C temp. gradient

End Box 1

Begin Box 2:
A widely-used microsensor node was developed through UCB’s DARPA project on SENSIT and Ubiquitous Computing. It is currently operational and produced in volume

by CrossBow (http://www.xbow.com). The core building block is a 1-inch x 1.5-inch "motherboard" comprising a low-power microcontroller (ATMEL 8535), low-power 900 MHz radio (RFM TR1000), non-volatile memory, LEDs, network programming support, and vertical expansion bus connector. The microcontroller contains FLASH program and SRAM data storage, ADC, and external I/O (standard I^{2}C and SPI, and direct ports). A second small microcontroller is present to allow the node to 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 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, and pressure, or actuator connections are 'stacked' like tiny PC104 boards. The processor dissipates several nano-Joules per 8-bit instruction.

The sensor board for the Berkeley platform consists of five different micro-sensor modules to support a wide variety of potential applications. The types of sensors it supports include: light, temperature, acceleration, magnetic field, and acoustic, and each of these sensors can be purchased off-the-shelf. All modules in the sensor board can be power cycled 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 I2C bus.



We have recently stacked a motor control board atop a micro-sensor node, and mounted the resulting assembly on a motorized chassis with two wheels [Sibley-02]. The motor control board regulates wheels speeds, and provides range information from two forward and one rear-looking infrared emitters. The resulting node (see Figure) is essentially a small mobile robot platform with the same network interface as the micro-sensor nodes described above.

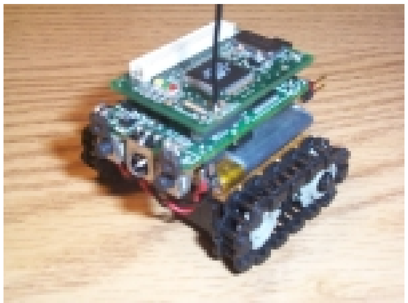


Figure: The Robomote: a mobile sensor node

End Box 2

B. Sensing and Actuation

Interfacing to the physical world involves exchanging energy between the embedded nodes and their environment. This takes two forms: sensing and actuation. Whatever the sensed quantity (e.g. temperature, light intensity), the sensor transducers a particular form of energy (e.g. heat, light) into information. Actuation allows a node to convert information into action. An actuator moves a part of itself, relocate spatially, or move other items in the environment (e.g. push an object or release a chemical). An important role for actuation is to enable better sensing. Sensing and actuation are together the means of physical interaction between the nodes and the world around them.

The decade of the 90s 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 of sub-millimeter mirrors that switch photons between different optical fibers. MEMS technologies are successful in the optical applications space because they provide far superior performance (TI DMD displays) or the same performance at lower prices (OMM Crossbar switches). MEMS technologies are successful in Detroit because they are reliable and dirt-cheap.

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

While 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 environment. These systems were unresponsive and slow to adapt to dynamic environments. Their disfunctionality in physical domains spurred the development of control techniques that closely coupled perception and action, with remarkable success. The earliest examples of these were stateless, reactive systems, which were responsive but lacked generality and the ability to store representation. Modern, behavior-based control [Mataric97] generalizes reactive control by introducing the notion of behavior – 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 [Arkin89], where a planner and a reactive system communicate through a third software layer designed explicitly for that purpose.

C. Localization

For a system to operate on input from the physical world, a key problem is for nodes to know their location in three spaces in order to provide *context aware* services to other elements of the system. A static or mobile node may answer the question ‘where am I?’ in several ways. The answer may be relative to a map, relative to other nodes, or in a global coordinate system. For a sensor network, this is particularly relevant, since the queries to which it will provide answers (based on sensor measurements) often need to be tagged with location information.

Localization plays a critical role in registration between the virtual and physical worlds. In a sensor network this can take a variety of forms; nodes in a network may report data tagged with relative location information, but from time to time, some nodes in the network may 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 in order to do so, it needs to know the baseline between the two nodes (i.e. the location of one relative to the other).

Scale and autonomy play an important role in location computation. Several early systems were large-scale but relied on careful offline calibration and surveying before node deployment to ensure reliable localization. Examples include the active badge and active bat systems (see Box 3). In robotics, on the other hand the focus was on techniques for localizing small numbers of robots [Thrun-00] autonomously, i.e., without relying on pre-surveyed maps, beacons, or receivers. A recent trend has been to investigate algorithms that successfully localize large-scale, networks of nodes autonomously.

Localization techniques for embedded devices can be taxonomized in several ways; by the sensors used, whether the localization is with reference to a map, whether the localization is done using external beacons etc. (see [Hightower01] and references therein for a summary). Broadly speaking localization may be viewed as a sensor-fusion problem. Given disparate sources of information about different aspects of node location (and pose), 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 (e.g. large numbers of small nodes), nodes cannot be placed with great care. In some systems nodes may be mobile (either because they are robotic or because they may be moved by other entities in the environment). 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 where significant effort has been expended upfront to instrument the environment for the *purpose* of localization.

There is a wide range of algorithms for localizing both fixed and mobile network nodes. One example recently demonstrated coarse localization [Dedeoglu00] of networks of mobile nodes, by allowing them to build a topological representation (a map) of their environment. Significantly, the system not only localizes the nodes, but also does so adaptively, by updating an internal sparse representation of (a changing) environment. We have also demonstrated an algorithm that makes use of the nodes themselves as landmarks thereby allowing us to determine the location of each node without reference to environmental features or a map. The algorithm [Howard01] constructs a `mesh' in which nodes are represented by point-masses and observed relationships between nodes are represented by springs. A relaxation algorithm is used to determine the lowest-energy state of the mesh, and thereby determine the most probable location of the nodes.

BEGIN Box 3

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 large numbers of nodes (scale), but otherwise differed significantly from current trends in that they had relatively fixed (not ad hoc or variable) system structure, and the application emphasis was on providing context and location to human applications, with very simple forms of autonomy relative to those anticipated in future systems.

The first steps in the direction of interacting with the physical world as a general computing concept were taken at the Xerox PARC and Olivetti (Now AT&T) Research Lab in Cambridge. The emphasis in that work was to build a range of many different sized devices that could occupy distinct niches in the ecosystem of interactions with information. Much of the emphasis of the work was on understanding how people might interact with such devices and how user interfaces could be constructed to utilize contextual information about the collection of devices. Thus, the primary connection with the physical world involved determining the location of each of the devices and thereby their geometric relationship to each other.

The Active Badge system, developed between 1989 and 1992, utilized an IR beacon carrying identity information from small mobile devices to IR sensors in the environment. Given a map of the physical space with sensors as landmarks, the short range and inability to pass through solid objects provided a convenient means of determining approximate location by proximity. The sensors were all connected by a low-bandwidth wired network to a centralized processing capability, which could maintain a representation of all the tagged objects and their interrelationships in order 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 very small, low-cost packages [Want99].

More recently, the Active Bat system used trilateration of ultrasonic beacons to provide accurate position determination in a physical space [Harter99]. Receivers are placed in a regular grid in the ceiling tiles. An RF beacon serves to inform a particular mobile device to emit a ultrasonic beacon and also to start a time-of-flight measurement at the receivers. Arrival times of the edge of the pulse are conveyed over a wired network from the receivers to a central PC, which can calculate the position of the specific device. The largest system deployed uses 720 receivers to cover an area of 1000m^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 advancement of the localization technology the use of a richer set of sensor modes and the integration of communication with the sensing and control process.

The Media lab and the Aware Home project have developed more direct means of sensing the interaction of people with their environment. These efforts include floor sensors to determine the position and movement of individuals (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 of 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 utilizing many such instrumented devices interfacing to various PDAs and mobile computers.
End box 3

D. Distributed system architecture

The trends of small form factor, sensor and actuator enabled, localizable, devices are key enablers. However, looking forward, it is the distributed system architecture that will make or break our ability to effectively instrument the physical world. Moreover, the same characteristics that enable these systems (e.g., miniaturization and wireless communication) impose constraints that require significant changes in the overall distributed system architecture in order 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. Added to this is the fact that wireless communications is the primary consumer of this energy in the context of low-power communication in physically complex settings [Pottie-Kaiser00]. Consequently, long-lived autonomous systems cannot be realized 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 time series data can be processed locally. Instead of building a network of sensors that all output high-bandwidth bit streams, we need to construct distributed systems whose outputs are at a higher semantic level, e.g., compact detection, identification, tracking, pattern matching, etc.

To support such an architecture, there are important systems oriented trends emerging. Two important examples are self-configuring networks and data-centric systems. One objective of such self-configuration is to allow systems to exploit node-redundancy (density) to achieve longer unattended lifetimes. Techniques for coordinating and adapting node sleep schedules to support a range of tradeoffs between fidelity, latency, and efficiency are emerging [Chen01**,** Cerpa02, Schurgers02, Xu01]. Ad hoc routing techniques represent an even earlier example of self-configuration in the presence of wireless links and node mobility [Johnson96] However, ad hoc routing still supports the traditional IP model of shipping data from one "edge: of the network to another. Small form factor wireless sensor nodes can on the one hand not afford to ship all data to the edges to have the processing done outside the system, and fortunately do not need to operate according to the same "layering restrictions" as IP networks when it comes to application-layer processing of data at "intermediate" hops. Directed diffusion promotes in-network processing by building on a data-centric, as opposed to an address-centric architecture for the distributed system/network. By using data naming as the lowest level of system organization, flexible and efficient in network processing can be supported; see Box 4 and [Heidemann01].

Begin Box 4

Directed Diffusion has been implemented on several platforms and 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 (PC-104s), as well as on Sensoria's platform (prior versions also ran under Windows CE on older Sensoria hardware). A constrained subset of Diffusion has also been implemented under TinyOS and runs on the UCB Motes (See Box 2). These implementation support a common API.

In Directed Diffusion sensors *publish* data and clients *subscribe* to data. Both identify data by attributes, e.g., "the southwest", "acoustic sensors". Diffusion divorces data identity from host 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:

- *Keys*: latitude, longitude, sensor-type, etc.
- *Values*: type (integer, blob, etc.) + data
- *Operations*:
- *formals: EQ, LT, EQ_ANY, etc. [conditionals]*
- *actual: IS [specifies data]*

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 distributed to the sensor network to perform application-specific, in-network processing tasks such as data aggregation, caching, and collaborative signal processing. See [Heidemann01] for more details.

The APIs have been published and used by Cornell, BAE, Xerox PARC, and Penn State 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 implementation.

End box 4

A second trend is the increased reliance on tiered architectures in which smaller numbered higher-end elements are used to 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. It will be desirable to introduce tiered architectures, where some system elements have greater capacity. In a tiered architecture, the smallest system elements can be used to achieve spatial diversity and short range sensing, while the computationally powerful elements implement more sophisticated and performance intensive processing functions, such as DSP, localization, and long-term storage. These higher tier resources may 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.

The increasing research focus on autonomous and scalable, physically coupled systems bodes well for continued progress towards Weiser's vision. We conclude with a description of some particularly critical systems challenges.

IV. Where are we headed: Systems challenges and concluding remarks

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 may mean wireless sensors for measurement and logging in every test tube and beaker. For the ubiquitous computing environment it may mean sprinkling sensing capabilities through a space appropriate to the activities of interest, rather than conducting studies in a specially designed testbed environment**.** To realize 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 require 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 a system wide architecture is the shift to naming data in terms of the relevant properties of the physical system instrumented, instead of in terms of the computing system embedded; effectively, naming data instead of naming nodes [Heidemann01, Adjie-Winoto99]. Many systems will be organized around spatial and temporal coordinates. Given a tuple-space for the instrumented environment, we require a programming model for the computations distributed in time and space. Various models are under consideration, which alternatively view the system as a distributed database [Bonnet00] or a loosely coupled parallel computing structure.

Increasingly physically embedded systems will need to self-organize. Self-organization, particularly *spatial* reconfiguration, will be needed for addressing variability at multiple scales. An interesting (and unique) aspect of the interaction with the physical world is the ability to manipulate it. While the systems of today are being built with self-configuration in mind, in the longer term these systems will integrate manipulation and reconfigure themselves and the world in which there 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 that these systems ultimately disappear into the environment … they must do so while operating correctly, and to the greatest extent possible, also when they fail. Users will not rely on these systems if they do not degrade with predictable behaviors and if they do not lend themselves to intuitive diagnosis, which comes with the ability to develop mental models [NRC-EE01].

Finally, another potential stumbling block is the potential for damage or **intrusion on the spaces** instrumented. Two rather different forms of this 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. However, ultimately many of these issues will require articulation of desired social and legal policy as well.

As the composition of the author list, and the brief description provided attests, this is an inherently and unavoidably multifaceted problem domain. It calls for the community to embrace interdisciplinary approaches and to transform the education 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 sub-discipline is in this area. Examples of the disciplines involved include networking (large-scale, distributed, wireless), OS (small footprint), databases (query processing for streaming data from real sensors), AI (robotics, vision graphics): virtual reality (instrumenting people with sensors and feeding them data based on these sensors), and signal processing (Collaborative signal processing [Zhao02]). One of the implications of the interdisciplinary nature of this area is the need for upgrading training of our students at both the undergraduate and graduate level. Students need skills that extend beyond constructing complex programs to manipulate virtual objects. Students need the ability to understand and model physical world processes which they will have to measure and with which they will have to cope. [NRC-EE01]

V. References

[Adjie-Winoto99] William Adjie-Winoto, Elliot Schwartz, hari Balakrishan, Jeremy Lilley, “The design and implementation of an intelentiional naming system”, SOSP 1999

[Arkin89] R. Arkin, “Towards the Unification of Navigational Planning and Reactive Control”, AAAI Spring Symposium, Stanford University

[Bonnet00] Philippe Bonnet, J. E. Gehrke, and Praveen Seshadri. Querying the Physical World. IEEE Personal Communications, Vol. 7, No. 5, October 2000, pages 10-15. Special Issue on Smart Spaces and Environments.

[Cerpa01] Alberto Cerpa, Jeremy Elson, Deborah Estrin, Lewis Girod, Michael Hamilton and Jerry Zhao, “Habitat monitoring: Application driver for wireless communications technology”, 2001 ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean, Costa Rica, April 2001.

[Cerpa02] Alberto Cerpa & Deborah Estrin. “ASCENT: Adaptive Self-Configuring sensor Network Topologies”. To appear in IEEE Infocom 2002, NYC.

[Chen01] Benjie Chen, Kyle Jamieson, Hari Balakrishnan, and Robert Morris. Span: an Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks.
Proc. 7th ACM MOBICOM Rome, Italy. July, 2001.

[Chu02] M. Chu, H. Haussecker, F. Zhao, "Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks." Int'l J. High Performance Computing Applications, to appear, 2002.

[Dedeoglu00] Dedeoglu G. and Sukhatme G.S. (2000) Landmark-based Matching Algorithm for Cooperative Mapping by Autonomous Robots*Proceedings of the 5th International Symposium on Distributed Autonomous Robotic Systems*, Knoxville, Tennesssee, October 4-6, pp 251-260

[Doherty01]
L. Doherty, B.A. Warneke, B.E. Boser, K.S.J. Pister,
"Energy and Performance Considerations for Smart Dust,"
International Journal of Parallel Distributed Systems
and Networks, Volume 4, Number 3, 2001, pp. 121-133.

[Estrin99] Deborah Estrin, Ramesh Govindan, John Heidemann and Satish Kumar "Next Century Challenges: Scalable Coordination in Sensor Networks", ACM MobiCom 99, August 99, Seattle, WA.

[Heidemann01] John Heidemann, Fabio Silva, Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, and Deepak Ganesan, “Building Efficient Wireless Sensor Networks with Low-Level Naming”, SOSP 2001

[Harter99] Harter, A., Hopper, A., Steggles, P., Ward, A., Webster, P., “The Anatomy of a Context-Aware Application., Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking, MOBICOM'99, Seattle, Washington, USA, August 1999, pp. 59-68.

[Hightower01] Jeffrey Hightower and Gaetano Borriello, "Location Systems for Ubiquitous Computing," *Computer*, vol. 34, no. 8, pp. 57-66, IEEE Computer Society Press, Aug. 2001

[Hill00] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, Kristofer Pister, “System Arechitecture Directions for Networked Sensors”, ASPLOS 2000

[Howard01] Howard, A., Mataric, M.J. and Sukhatme, G.S. (2001) "Relaxation on a mesh: A formalism for generalized localization", *2001 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 29 - November 3, Maui

[Johnson96] Johnson, D., Maltz, D., “Protocols for Adaptive Wireless and Mobile Networking”, IEEE Personal Communications Magazine, February 1996.

[Kahn00]
J.M. Kahn, R.H. Katz, and K.S.J. Pister,
"Emerging Challenges: Mobile Networking for Smart Dust",
J. of Commun. and Networks, vol. 2, no. 3, pp. 188-196, September 2000
http://www.eecs.berkeley.edu/~jmk/pubs/jcn.00.pdf

[Kidd99] Kidd, Cory D., Robert J. Orr, Gregory D. Abowd, Christopher G. Atkeson, Irfan A. Essa, Blair MacIntyre, Elizabeth Mynatt, Thad E. Starner and Wendy Newstetter.

"The Aware Home: A Living Laboratory for Ubiquitous Computing Research"
In the Proceedings of the Second International Workshop on Cooperative Buildings, *CoBuild'99*

[Kindberg01] Tim Kindberg & John Barton. A Web-Based Nomadic Computing System. In Computer Networks, Elsevier, vol 35, no. 4, March 2001, pp. 443-456.

[Li02] "Detection, Classification and Tracking of Targets in Distributed Sensor Networks", Dan Li, Kerry Wong, Yu Hen Hu, Akbar Sayeed
IEEE Magazine on Signal processing Special issue on wireless sensor networks, To appear 2002

[Mataric97] M. J. Mataric, " Behavior-Based Control: Examples from Navigation, Learning, and Group Behavior", *Journal of Experimental and Theoretical Artificial Intelligence,* special issue on Software Architectures for Physical Agents, 9(2-3), H. Hexmoor, I. Horswill, and D. Kortenkamp, eds., 1997, 323-336.

[Mercury] http://crl.research.compaq.com/projects/mercury/

[MITMediaLab]
http://www.media.mit.edu/context/
http://www.media.mit.edu/physics/
http://www.media.mit.edu/resenv/

[***NAP97] Ref Energy-Efficient Technologies for the Dismounted Soldier (1997)
http://books.nap.edu/books/0309059348/html/index.html

[NRC-EE01] Embedded Everywhere: A Research Agenda for Networked Systems of Embedded Computers, CSTB, NRC, 2001.

[Pister99] K.S.J Pister, J.M.K., B.E. Boser, *Smart Dust: Wireless Networks of Millimeter-Scale Sensor Nodes.* Electronics Research Laboratory Research Summary, 1999

[PocketPCCamera] http://www.hp.com/jornada/products/520/acc_pocket_camera.html

[Pottie-Kaiser00] [20] G. Pottie and W. Kaiser. Wireless Integrated Network Sensors. *Communications of the ACM*, 43 (5): 51-58, May 2000.

[Pradhan02] "Distributed Compression in a Dense Sensor Network", S. Sandeep Pradhan, Julius Kusuma, Kannan Ramchandran
IEEE Magazine on Signal processing Special issue on wireless sensor networks, To appear 2002

[Schurgers-02] C. Schurgers, V. Tsiatsis, M. Srivastava, "STEM topology management for efficient sensor networks", To appear in IEEE Aerospace Conference, Big Sky MT, March 2002.

[Sibley-02] Gabriel T. Sibley, Mohammad H. Rahimi and Gaurav S. Sukhatme, “Robomote: A Tiny Mobile Robot Platform for Large-Scale Sensor Networks”, Proceedings of the IEEE International Conference on Robotics and Automation, May 2002, Washington DC. (accepted, to appear)

[Thrun-00] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 101:99-141, 2000.

[Want99] R. Want, K. P. Fishkin, A. Gujar, B. L. Harrison, “Bridging Physical and Virtual Worlds with Electronic Tags", CHI 99

[Want92] Roy Want, Andy Hopper, Veronica Falcao, Jonathon Gibbons,
The Active Badge Location System. ACM Transactions on Information Systems, Vol. 10, No. 1, January 1992, pp 91-102. Technical Report 92.1

[Warneke01]
B. Warneke, M. Last, B. Leibowitz, K.S.J. Pister,
"Smart Dust: Communicating with a Cubic-Millimeter Computer",
Computer Magazine, Jan. 2001. IEEE, Piscataway, NJ. pp. 44-51
http://ieeexplore.ieee.org/iel5/2/19363/00895117.pdf

[Zhao02] Feng Zhao, Jaewon Shin, James Reich "Information-Driven Dynamic Sensor Collaboration for Target Tracking", http://www.parc.xerox.com/spl/projects/cosense/pub/ieee_spm.pdf
IEEE Magazine on Signal processing Special issue on wireless sensor networks, To appear 2002 [ORL] http://www.uk.research.att.com/bat/

[Xu01] Xu, Y,, Heidemann, J., Estrin, D., “Geography-informed Energy Conservation for Ad Hoc Routing”, In Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking(ACM MobiCom), Rome, Italy, July 16-21, 2001.