

Association for Information Systems AIS Electronic Library (AISeL)

AMCIS 2004 Proceedings

Americas Conference on Information Systems
(AMCIS)

December 2004

Wireless Sensor Network Applications

Mark Gaynor
Boston University

Mark Gaynor
Boston University

Matt Welsh
Harvard University

Austin Rowan
Boston University

Edward LaCombe
Boston University

See next page for additional authors

Follow this and additional works at: <http://aisel.aisnet.org/amcis2004>

Recommended Citation

Gaynor, Mark; Gaynor, Mark; Welsh, Matt; Rowan, Austin; LaCombe, Edward; and Wynne, John, "Wireless Sensor Network Applications" (2004). *AMCIS 2004 Proceedings*. 206.
<http://aisel.aisnet.org/amcis2004/206>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2004 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Authors

Mark Gaynor, Mark Gaynor, Matt Welsh, Austin Rowan, Edward LaCombe, and John Wynne

Wireless Sensor Network Applications

Mark Gaynor
Boston University
mgaynor@bu.edu

Steve Moulton
Boston University
smoulton@bu.edu

Matt Welsh
Harvard University
mdw@eecs.harvard.edu

Austin Rowan
Boston University
arowan@bu.edu

Edward LaCombe
Boston University
elacombe@bu.edu

John Wynne
Boston University
jwynne@bu.edu

ABSTRACT

This paper discusses applications of wireless sensor networks along with the challenges of integrating real-time sensor data with traditional IT environments. Two widely different sensor network applications are discussed: emergency medical services and supply chain management, along with the challenges of integrating sensor technology into current IT environments. Technical challenges center on the development of the sensors and sensor network infrastructure and include the need to comply with emerging application program interfaces (APIs) for grid/web services. Process-driven challenges, which center on the development and adoption of new business models and new applications, are the actual drivers for this technology. Both of these applications acknowledge the nearly limitless potential of sensing and connecting our physical world with wireless sensors¹ and help illustrate the difficulties of integrating sensor data into current IT infrastructure.

Keywords

Sensors, Sensor applications, wireless ad-hoc networks

INTRODUCTION

Small, low-power wireless sensor networks allow the physical environment to be monitored at high resolution and will lead to a vast increase in the quantity and quality of data available to organizations. A diverse array of monitoring applications is envisioned: environmental, buildings and bridges, perishable foods and fragile goods, as well as medical patients.

We are developing two unique sensor network application domains: patient monitoring and supply-chain management. For the former we built an emergency medical triage application that uses wireless pulse oximeters: tiny sensors that measure heart rate and blood oxygen levels. This application allows vital signs from multiple patients to be relayed to handheld PDAs carried by pre-hospital emergency medical technicians (EMTs) and paramedics. In the latter application, Sears Canada is working with our research team to devise the next generation smart warehouse and smart product management. Sears also believes that they can improve customer service by sensing that a shipment may be damaged before delivery to the customer, as well as the ability to sense when an appliance is about to break. Sears is also interested in tracking custom products, which is inefficient in their current infrastructure. The smart sensor application will provide better customer service at a lower cost. Both the medical and supply chain applications have similar requirements: integrating existing work processes with data generated by (potentially large numbers of) wireless sensors.

This section ends with a brief introduction to wireless sensors. The next section discusses two application areas (medical and product tracking) and presents several examples of applications we are beginning to build. Section 3 discusses how

¹ This work is sponsored by NSF grants (ACI-0330244,PFI-0227879)

applications need an API based on SOAP/XML (web/grid services [1]). We also explore other difficulties of this integration, such as energy and bandwidth conservation, without limiting the flexibility of application developers.

Wireless Sensors

The emerging domain of wireless sensor networks has yielded devices with a wide range of computation, communication, and sensing capabilities. These range from relatively powerful systems with PC-class processors and high-bandwidth wireless interfaces (e.g. IEEE 802.11), to much smaller, low-power nodes consisting of simple embedded microcontrollers and low-bandwidth radios operating in the 433 or 916 MHz ISM bands. Other wireless technologies, such as IEEE 802.15.4 (Zigbee), Bluetooth, and long-distance cellular have been investigated for sensor networks as well.

The core challenge spanning these platforms is managing the tradeoff between local computation and communication. Wireless sensor networks are typically battery-powered and therefore have a fixed energy budget to meet a given application lifetime. Moreover, the energy cost to transmit even small amounts of data greatly dominates that of computation [2]. Therefore, it is often necessary for sensors to expend significant CPU cycles to perform local compression, filtering, and aggregation of data in order to save communication overhead.

In this paper, we focus on the MICA mote, developed at UC Berkeley in collaboration with Intel Research (Figure 1). This device consists of an 8-bit Atmel ATMEGA 128L microcontroller, 132K of memory, 512K of nonvolatile flash memory, and a 19.2 Kpbs radio operating in the 433 MHz or 916 MHz spectrum. The device consumes approximately 25 mA with both the CPU and radio active, limiting lifetime on 2 AA batteries to about 5-6 days without any power-saving techniques. Lifetime can be extended significantly, up to months or years, by selectively powering down the radio, CPU, and other peripherals and operating at low duty cycle. A wide range of sensors, including light, temperature, magnetometers, accelerometers, humidity, and passive infrared, have been integrated with the MICA device. The MICA node is programmed in NesC [3], a component-oriented variant of C, and uses the TinyOS [4] operating system, a very lightweight OS developed specifically for this platform.

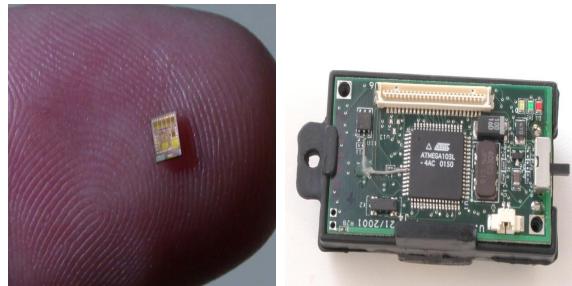


Figure 1 – Example of Motes

APPLICATIONS

There are many applications of wireless sensors, in many areas, from military and homeland security, to medical, to running more efficient businesses. The new breed of smart sensors provides tremendous flexibility to applications developers. Sensors can operate disconnected from any Internet while they sense, filter, and process local information. Smart sensors can add context to sensed data; for example, depending on what the item is, sensors can determine if it is broken or spoiled. Intelligent sensors enable a more dynamic and scalable architecture for many applications.

Business Case

There are good business reasons for using intelligent sensors to move processing to the edge of a sensing network, which follows, to some extent, the end-to-end argument of dumb networks with smart end points. Sensors with built in intelligence have many advantages over dumb sensors: they can store data, they have context for the data being collected, they work in disconnected operation, and they allow more scalable architecture for some applications. Being contextually aware of the environment enables more dynamic applications. For example, theft detection with a smart sensor is much more dynamic because the sensor can decide whether the product to which it is attached is being stolen and then report this fact. This type of

event driven architecture with smart sensors is far more flexible than dumb sensors that report a value when polled. Intelligent sensors will promote experimentation because of their ability to be re-tasked and enhanced programmatically²

One application of smart sensors does not involve any sensing: using a mote as a dynamic header. This idea, advanced by John Henderson, uses memory, communications, and processing power of a mote to write, save, and retrieve data dynamically on a mote. We are using the idea in both applications we are working on: In our emergency medical triage application, the smart sensors that collect vital sign information can also record the history of a patient; in our smart warehouse application, we program information about customization and history on a mote attached to a item in the warehouse. A dynamic header is more flexible than static headers such as bar codes, and RFID tags, which have a fixed header they transmit when sensed.

One advantage of intelligent distributed systems is the ability to build an overlay system on top of a current infrastructure. Many organizations have IT systems based on bar codes or RFID tags. It is relatively inexpensive to build a wireless grid of sensors and tag appropriate items with smart sensors. The new system runs on top of the current system, not replacing, but enhancing the operation. This overlay architecture is important to business because it enables capturing the value of the new sensor technology without losing the investment in the current infrastructure. By using smart sensors sparingly, for example on expensive and fragile plasma TVs, goods that spoil and cause health problems such as seafood, or goods that lose value when exposed to temperature extremes such as expensive wine, or custom orders that are hard to track with static systems based on bar codes or RFID tags, organizations capture value by focusing the new technology on areas with the most potential for gains in efficient operations. IT systems based on smart sensors will co-exist with current IT systems based on more traditional technology.

While supply chain management and emergency medical services may seem very different, they have many of the same requirements. Tracking is one common task; either a patient or item might need to be located. Customization is another common task; a patient has a unique set of procedures executed on them as well as subsequent reactions to these procedures, while a custom product has a unique set of customizations. These applications need to sense change: is a pallet of goods being stolen from a truck or warehouse, was a fragile item dropped, or did the condition of a patient change? We have found that our applications share many requirements.

Medical

A variety of vital sign sensor network architectures have been proposed. These range from intelligent, wireless sensors linked by a Personal Area Network (PAN) [5] to tightly integrated sensors on a single mote platform [7]. The latter appears to be more flexible and a better choice for short and long-term monitoring, due to the simplicity of wearing a single, lightweight device. The device platform should accommodate a range of sensors, depending on need, expendability, and cost. The simplest device would be composed of a mote, a small battery, and a pulse oximetry module. It would monitor heart rate and percent oxygen saturation of the blood. It could be used in many types of healthcare and homecare settings to monitor a variety of cardiopulmonary diseases, such as congestive heart failure, COPD (chronic obstructive pulmonary disease), asthma, and sleep apnea. Sensor data could be filtered and stored on the mote by a variety of programmable sensor algorithms, or the data could be filtered and sent wirelessly to a PDA, smart phone, or wearable computer. These more powerful, programmable devices would act as intermediaries between individual patient sensor nodes and the Internet. They would provide an alarm function and permit the sensor data to be viewed locally, stored, or sent via the Internet to an off-site base station for continuous real-time or archival healthcare monitoring.

The tight integration of additional sensors on a single mote platform, such as EKG (electrocardiography), respiratory movement, and non-invasive near-infrared determination of blood glucose, would optimize the power efficiency of the devices and enable more intensive physiologic monitoring of acutely ill and injured patients. Similarly, when a multi-sensor node is monitoring a relatively stable patient, only one or two of the sensors should be “on,” in order to preserve battery life. Smart sensors would therefore monitor their output and, in the event that certain alarm parameters are exceeded, turn on additional sensors to help determine whether the alarm is real or not.

This type of smart sensor would have great utility in many patient care settings, especially those in which wired monitors are cumbersome and patient vital signs tend to vary widely. An ideal application would be in the pre-hospital phase of trauma care. Here, vital sign information could be collected by the mote and automatically uploaded to a mobile computer on which a paramedic or EMT is capturing additional patient information. The integrated vital sign and patient care data could then be

² This is similar to what happened with the PBX [5].

relayed ahead of the patient to a hospital or, in the case of a mass casualty event, networked and sent to a base station or command center where aggregated sensor and patient care data is sorted, summarized, and displayed in a graphical format to provide a dynamic, global view of the evolving situation.

Ad-hoc Medical Sensor Networks

Mass casualty events (MCEs) are well suited to the application of ad-hoc sensor networks, for they are sudden, unexpected situations in which, over a short period of time, large numbers of casualties are generated and organized community support mechanisms are either crippled or destroyed [8]. A central problem is the large number of casualties and how to both monitor and deal with them. If the injured and severely injured cannot be triaged in a rapid and coordinated manner, their large numbers can quickly overwhelm and prevent emergency field personnel and hospital staff from providing quality trauma care [9]. Application of ad-hoc sensor networks to mass casualty events could help mitigate the discrepancy between patient load and available resources. The sensors themselves could be used to identify, monitor, and help separate those who will do well with minimal care from those who will die despite maximal care, so that focus can be directed to those with treatable, life threatening injuries. It is in this latter group that careful planning and coordination of care can make the biggest difference.

Sensor information needs to be linked with patient information that is collected by on-scene field personnel. Aggregate patient data would then be sent to a central command center where a global view of the entire scenario would unfold (Figure 2 and Figure 3). Triage information would then be relayed back to field personnel, who transport victims to pre-arranged hospitals, according to known injury severity and medical center capabilities.

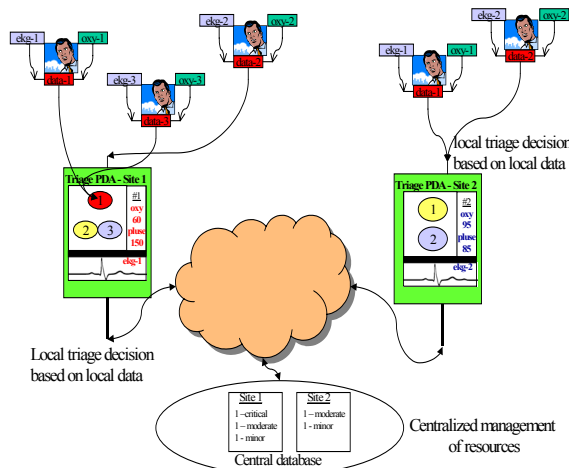


Figure 2 – Triage System Architecture

Decisions regarding who to transport and where are significantly influenced by accurate and timely information. In current wireless sensor networks, the communication model assumes that all nodes should be given a fair share of the radio medium; nodes are therefore extremely conservative about transmitting data when other nodes may be using the channel. This fairness assumption also limits communication bandwidth: the current MICA platform imposes a per-packet back off delay of up to 50 packet times to avoid colliding with another node's transmission. One solution is time division multiple access (TDMA), which allows nodes to communicate within a fixed time slot where they are guaranteed to avoid collisions with other nodes. However, this approach requires nodes to be time synchronized and does not in itself address the prioritization issue. It will therefore be important to develop a medium access control scheme for critical scenarios that suppresses communications from non-critical nodes and devotes greater channel resources to relaying critical data.



Figure 3 – Prototype Local Triage Application

Handing Off Sensor Nodes

In time, a single field provider could be monitoring and providing care for several patients at a time. The patients, and thus the nodes with their stored patient information and vital sign data, must be able to be handed off to newly arriving field providers who will transport patients to area hospitals. These individuals may be using different resuscitation software on their wearable computers, and the motes should be able to negotiate the hand-off of their information in a smooth and error-free manner. This type of contextual, interoperability will be important if the use of these sensors is to become widespread.

Embedded Medical Sensor Networks

Data capture with a multi-sensor node applied in the field could continue in the hospital and the operating room to continuously capture and integrate key vital sign information with accruing patient information during the critical phases of early hospital care. In time, broader application of the multi-sensor devices for monitoring during major elective surgical procedures and in the intensive care unit could be used to guide fluid therapy for major medical and surgical illnesses (e.g. sepsis, post-MI cardiogenic shock, and non-operative trauma). This information would allow physicians to monitor in real-time how various changes and types of fluid therapy affect patient status and outcome. As data is accrued, algorithms will be developed to accurately guide and control fluid resuscitation and drug administration. Eventually, automated resuscitation algorithms will be developed to enable computer controlled, closed loop infusion and drug therapy protocols that can be titrated to established endpoints.

Warehouse

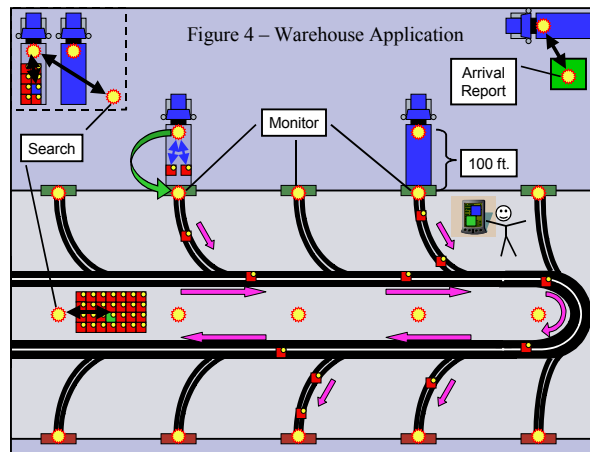
The Boston University Leading in the Dynamic Economy (BUILDE) agreed in the summer of 2003 to pursue a research project in conjunction with Sears Canada exploring the potential business value of using Berkeley motes for local data management and asset tracking in Sears' supply chain. The goal of this project, completed in September 2003, was to demonstrate that user defined data could be reliably written to and read from non-volatile memory on board the motes for the purpose of locating product and supply chain data directly on a piece of merchandise. Our simple prototype stores alphanumeric user-defined product data on the mote to illustrate how Sears can create value with this technology. We also created a database to track the information on the mote.

The motivation for the development of this experiment was twofold. Although Sears maintains a world-class logistics operation and has highly advanced IT systems in place in its supply chain, sometimes Sears' best customers do not get the best service. This is because the systems in place are sometimes inefficient at handling products which have been customized in some way, for instance high-end furniture with custom upholstery. Current systems also have difficulty tracking whether pieces of merchandise, which are parts of a set and should be shipped together (i.e. the pieces of a living room set), have all arrived and are ready to be shipped out as a unit.

The second motivation for the pursuit of this experiment is the desire to locate sensing capabilities on products which are very expensive and fragile, such as a plasma screen television or high-end appliances with advanced electronics. Later phases of this experiment will include outfitting motes with sensors that can detect whether an item has been dropped, and if so, where and when it happened and how serious the impact was.

Figure 4 illustrates a simple prototype of the application we are developing. Items or pallets of items have smart sensors embedded within them and these are sensed as the truck on which they are loaded passes the entry gate. These trucks are either stacked up (upper left hand corner) or sent to an unloading dock. If stacked up, the trucks may be searched for certain items. When unloading the truck, each item is sensed as it moves into the warehouse on the central conveyer belt. Items are

stored or directly sent to the outgoing shipping dock. This prototype design will track items from first contact to final exit, thereby allowing more dynamic supply chain management.



In general, the motes do not have context, nor do they assign meaning to their data. The write functionality of the application has three phases. The user uses a Java tool to enter a mote ID, a line number, and an alphanumeric value and issues a write command. This value is then written to the specified line of the specified mote. That value is then instantly read back out (as a check) and transmitted back to the PC as it is written to a database resident on the PC. The database tables, which then contain an image of the data on the mote, provide context for that data and a convenient platform for viewing it.

Our field-test of this application at Sears involved tagging actual pieces of merchandise with motes encased in sturdy plastic enclosures. We then followed these items through a large distribution center as they went through each phase of the receiving, staging, and shipping processes, and we wrote various pieces of product and shipping information to the motes. Data logged included product information such as Product Description Manufacturer Date/time shipped from Manufacturer, Customer Name, and the like. The application performed very reliably in the warehouse setting we worked in, and transmission range proved to be more than adequate, as we were able to reliably read and write at a 50-foot distance. This application has been ported to a small handheld device (Microsoft PocketPC platform) and was able to automatically detect, read and update motes in field tests at a Sears distribution center. Later efforts will focus on ways to drastically reduce the amount of manual data entry necessary for capturing desired information.

INTEGRATING SENSORS INTO IT APPLICATIONS

There are two major challenges to integrating wireless sensors into today's IT environment: 1) providing a standard API to application programmers (one that is familiar and supported by their development environment), and 2) the technical difficulty of building a sensor network infrastructure that allows robust, scalable, and secure applications that use sensor data, while still allowing experimentation within both the sensor and application layers.

Standard API to Applications

For widespread adoption of wireless sensor technology application, developers must have access to sensor data within their development environment. One popular API is based on XML, SOAP (and maybe OGSA grid standards). Currently, all major vendors offer development tools supporting this infrastructure. Microsoft promotes its .NET framework. Sun is behind Java based J2EE (Sun ONE), and IBM supports its Dynamic e-Business initiative. There are also open source web services development environments, such as Axis in the Apache framework [10]. All of these different development environments allow building web/grid services, which are usable by any client, on any host, with any operating system, as long as these web service standards are complied with. For now, pervasive adoption of wireless sensors depends on their data being encoded in XML and packed within a SOAP envelope.

In order for applications to use sensor data, these applications must discover and understand how to use distributed services based on the web services. This means that services based on sensor data must be described with Web Services Description Service (WSDL) and discoverable using WS-Inspection (the distributed version of Universal Description, Discovery, and Integration (UDDI) [11]). When services based on sensor data are encoded using XML, transported in a SOAP envelope,

described with WSDL, and registered in a directory, then applications will have an API supported by their new web services development environments.

Web services define a standard for the type of services and their operations. These services range from querying the service for data, and registering a service to allow dynamic discovery, subscribing to event notification. Sensor services that adhere to the basic web services stand the best chance of rapid adoption because they fit into the strategy of major vendors such as Microsoft, Sun, and IBM.

Technical challenges

There are many technical challenges to integrating wireless sensors within IT environments. These include integrating non-IP devices into the Internet, no unique global addressing scheme that disable end-to-end applications, disconnected operation, non-reliable MAC layer protocols, designing efficient routing protocols, and difficult programming environments. Because these small self-contained sensors often use batteries for their power, there are severe power constraints on applications; it follows that encoding sensor data into XML and then encasing this data in a SOAP envelope is expensive both in terms of central processing unit (CPU) cycles and the number of bits transmitted across the wireless link. This suggests that one main challenge with integrating these low power sensors is the conversion of sensor data into the web services architecture. Included in this challenge is converting data into XML (based on a schema) within a SOAP envelope and then using standard Internet protocols such as (SMTP, HTTP, FTP, ...) to transport this data to applications.

Sensors do not have a unique global addresses, which implies that end-to-end applications do not work. Within a particular sensor realm, however, each sensor will have a unique address. For end-to-end applications a mapping is needed to allow applications to access (and identify) sensor data. It is not clear what end-2-end connectivity means in the sensor world.

Resource Allocation

Power management to extend battery life is a prime consideration in many sensor applications. Power management defines many characteristics of a sensor node: how often does it listen for incoming messages, how often does it wake up to send a message, how are CPU cycles traded for the number of bits transmitted, addressing scheme, and protocols at the MAC and network layers. These parameters vary depending on the application space; for example, one expects to sample vital signs of an emergency patient more frequently than one would measure the temperature of a shipment of wine.

Since processing data requires CPU cycles, moving processing within the network seems like a good idea, and it can be. The tradeoffs are complex, however, because it is very expensive to send bits over the radio [2], which implies that trading some local processing for reduced bandwidth is wise. Some processing, such as XML encoding of binary (and perhaps compressed) sensor data, is obviously a good choice as an in-network service because if this encoding was preformed on-sensor, it would increase both CPU cycles and required BW. For other choices, such as monitoring many sensors for abnormal conditions (such as temperature extremes and drop detection), it makes sense to perform these on the sensor because a few CPU cycles significantly reduces the required BW of the applications. This is particularly true when the abnormal condition being sensed is relatively rare. Some choices are not clear, however, such as filtering out small movements when tracking location. Moving services from the sensors to the network can improve the longevity of battery life in sensors; however, the choice of what function to perform where is highly application dependent.

There are other technical problems with providing in-network services; historic evidence from the telephone and Internet [5] indicate that it is very difficult to predict what services will be successful, as well as what feature sets these services should have. This same evidence also indicates that in-network services make sense in many cases, particularly when the uncertainty of what users desire in these services is low. When user uncertainty is high, then the value of allowing end users to experiment often exceeds the gains in efficiency from in-service services [5,12]. Perhaps most important is to have a flexible environment in the context of in-networks services. During the experimental phase, end users should be allowed to use only minimal in-network services; however, when users have selected the best results from this experimental phase, in-network services should be able to provide a more scalable, efficient, and robust application. During the experimental phase, it is possible that in-network services need to evolve to support newly discovered requirements. Building a sensor network architecture that does not stifle user innovation by constraining them with intelligent network services, while allowing selected successful applications to be implemented with intelligence in-network services, is an important research question.

Routing within sensor nets is hard because of limited power and the dynamic nature of sensor networks. The low bandwidth and power makes traditional Internet routing protocols (such as RIP (bandwidth inefficient), and OSPF (too complex) too expensive. Sensor nets also have the potential to be much more dynamic than the current Internet because small sensors are

very mobile and likely to be on people, cars, and other moving objects. One open research question is how to route in such a power constrained, yet dynamic, environment. Routing strategies, such as GPRS, address non-unique addresses between different sensor regions. Other routing algorithms, such as data-centric [13], present a framework based on data aggregation, rather than routing, which is based on the shortest path. Geographic [14] routing might work well when sensors have built in GPS systems. The point is that routing within a region of a sensor network is specialized, complex, and dependent on the ability of the sensor nodes.

CONCLUSION

Emerging technology supports the development of wireless sensors and sensor networks to assist in the management of pre-hospital and hospital-based patient information. This information can be used to coordinate the linkage of pre-hospital transport systems with their hospital-based resources; it can also be used to develop and refine additional tools for resuscitation and decision support.

We are working with Sears to develop strategies for successfully capturing value from wireless sensor technology. We believe that adding sensing capabilities to Sears supply chain will allow logistics managers to better track and manage items in their supply chain because they act as their own exception handlers, emitting appropriate and valuable notifications based on sensor events contextualized by business rules resident on the mote.

The integration of wireless sensor networks with the traditional wired grid will conceivably allow our physical world to connect with a nearly limitless variety of web-based information utilities and computation services. The end result could be a network infrastructure that supports, yet pervades everyday life in ways still unimaginable.

REFERENCES

1. Foster, I., C. Kesselman, et al. (2003). *The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration*.
2. Hill, Jason, [System Architecture for Networked Sensors](#) Ph.D. Thesis, UC Berkeley, May 2003.
3. David Gay, Phil Levis, Rob Behren, Matt Welsh, Eric Brewer, and David Culler, PLDI'03. [The nesC Language: A Holistic Approach to Networked Embedded Systems](#)
4. Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, and Kristofer Pister, ASPLOS'00. [System architecture directions for networked sensors](#)
5. Gaynor, M, *Network Service Investment Guild: Maximizing ROI in uncertainty markets*, Wiley, 2003.
6. Richard, Block, Weiser RR. First clinical trial with a telemetric shunt-integrated ICP-sensor. *Neurologic Research* 21:117-20, 1999.
7. Jovanov Raskovic , Price , Chapman J, Moore A, Drishnamurthy A. Patient monitoring using personal area networks of wireless intelligent sensors. *Biomedical Sciences Instrumentation* 37:373-8, 2001.
8. Welsh M, Myung D, Gaynor M, Moulton S. Resuscitation monitoring with a wireless sensor network. *Circulation* 108:1037 Supplement IV
9. American College of Surgeons Committee on Trauma 1998. *Mass casualties. Resources for Optimal Care of the Injured Patient 1999*. Chicago, IL: American College of Surgeons, 87-91.
10. Apache "<http://www.apache.org/>," The Apache Software Foundation, 2002.
11. Graham, Simeonov, Boubez, Davis, Daniels, Nakamura,, and Neyama, *Building Web Services with Java: Making Sense of XML, SOAP, WSDL and UDDI* Sams, Indianapolis, 2002, p. 581.
12. Gaynor, M. and Bradner, S. Using Real Options to Value Modularity in Standards.2001) *Knowledge Technology and Policy*, Special issue on IT Standardization, 14:2.*
13. Krishnamachari, Bhaskar, and Estrin, Deborah, and Wicker, Steven, "Modelling Data-Centric Routing in Wireless Sensor Networks" , *IEEE infocom* 2002
14. Karp, B., *Challenges in Geographic Routing: Sparse Networks, Obstacles, and Traffic Provisioning*, in the *DIMACS Workshop on Pervasive Networking*, Piscataway, NJ, May, 2001.