On Sensor Frameworks for Pervasive Systems

Billibon Yoshimi
IBM T. J. Watson Research Lab
P.O. Box 218
Yorktown Heights, NY 10598 USA
+1 914 945 3611
byoshimi@us.ibm.com

ABSTRACT

We believe an important part of any pervasive or wearable computing environment is an infrastructure for selecting and employing various sensors into an effective pervasive system. Without a uniform framework, researchers need to design their own ad-hoc structures and there is no common ground for system comparison and evaluation. Thus, our focus is on formulating a framework that can be used to support pervasive computing using sensors systematically. In this paper in particular, we address some of the fundamental issues in applying sensing and tracking to pervasive environments.

INTRODUCTION

While still in the formative stage, at IBM we are developing an architecture to support the needs of a pervasive system. The four main components of our architecture are: User Modeling, Data Store, Context Monitoring, and Ubiquitous Display. This paper concentrates on some of the problems and architectural issues associated with integrating sensors in such a system. In addition, we specifically propose integrating sensors in the User Modeling and Context Monitoring modules.

The idea of using sensors in a pervasive environment is not new. Xerox PARC [9] generated one of the first “ubiquitous” computing environments. It consisted of active badges and instrumented offices connected by a wired and wireless networking infrastructure. It also contained an infrastructure for tracking users, detecting and analyzing situations and performing actions based on those situations. One of the largest active badge efforts was deployed at Olivetti Research[4]. Georgia Tech[1] has made an initial attempt at defining context awareness in pervasive environments. More recently KTH[2], University of Washington[8], and UMTS (the next generation GSM)[5] have started pervasive efforts. A common thread among these research projects is tracking users and using tracked data to change to the environment (the environment includes computer displays, databases, and the user's perception of the world.) While all of these systems employ sensors in a fundamental way, none of these systems make use of a systematic framework for integrating sensors.

To further our understanding of how to apply sensors to these environments, we will begin by examining different sensor paradigms. We examine the concept of location and situation awareness. We will then examine problems related to integrating sensors with location and situation awareness. Finally we will describe our initial prototype and future directions for our research.

UNDERSTANDING SENSORS

Many system designers make the mistake of leaving the task of choosing sensors to the end of the system design process. Knowing sensor properties and developing a plan for examining and integrating sensors is a very important part of any system design process. To understand and model a wide variety of sensors, we begin by characterizing sensor by their functions. Based on the characteristics of different sensors, we propose several design considerations useful for system designers.

We define sensors simply as data producers. They may have a physical embodiment such as a camera or smart badge or they may be entirely logical entities like timer events. In their most rudimentary form, a physical sensor converts physical quantities into a computer understandable format. While most sensors include some signal conditioning circuitry, noise and sensor anomalies make most sensors finicky input devices. Sensor outputs are generally only useful to specific systems that have a straightforward connection to the sensed input. For example, thermostats control HVAC systems and ignition sensors control automobile idling. If the mapping is not straightforward, as is the case when using cameras or motion detectors to track humans, then a sophisticated layer of algorithmic processing is required before meaningful data can be generated.

Active vs. passive sensor systems

There are two major types of sensor. *Active sensor systems* interact with the environment and observe how their actions affect the environment. Examples of active systems include RF (radio frequency) transmitters including RFID tags (radio frequency identification tags), IR (infrared) transmitters, structured lighting, touch and most sonar and radars. *Passive sensor systems* sense ambient radiation or signals. Passive systems include GPS (Global Positioning Systems), ambient audio, most shape-from-X vision systems and passive motion detectors. The definition of ambient radiation can be extended to include satellite signals since some like GPS transmissions are not under the control of the sensor system.

Active systems work by actively controlling a probe signal in the environment and observing how interacting with the environment causes sensible changes. These changes can be compared with known observations to construct a model of the environment. Actively probing the environment helps to remove ambiguities from a passive sensor view. Because these systems irradiate their environment, they leave a detectable signature on the environment.

In contrast, passive systems simply receive information passively. Passive systems are useful in situations where irradiating the environment is undesirable or impossible. Locations, positions and shapes can be recovered from sensing the environment passively. Passive systems rely on the recovery of features or landmarks to distinguish between situations. Passive systems are at the mercy of their environments. If GPS satellites are out of view, or the lights go out in a building, passive systems must rely on other sensors to determine system status.

Active and passive systems also handle privacy differently. In most cases, personal privacy is a key concern in pervasive systems. Systems compromise privacy through two channels: sensing and communications. Active systems can give up position, location, and numerous other system parameters by actively interrogating the environment. Passive systems, on the other hand, require than an external source initiate sensing or communication. Through passive listening, the passive system gains information without giving up location and/or situation information. In short, system designers should construct sensing systems with a user's privacy needs in mind.

A discussion of sensing techniques is not complete without a discussion of sensor fusion. After selecting a suite of sensors to accomplish a given task, we must establish a data structure for fusing multiple sensor views together to form a unified view of the world. The multiple redundant readings give the system confidence to remove outliers and to make prediction based on faith in the veracity of the sensor reading. The fusion of data from multiple sources is a difficult task since each data stream is not guaranteed to have the same sample rate, baseline, or accuracy. Since every sensor is different, each sensor can return erroneous data. It is the job of the fusion engine to unify these semi-erroneous results into a coherent picture of the world. While several methods exist for fusing the results of multiple sensors, configuring a sensor fusion subsystem often requires expertise in control theory, probabilistic reasoning or knowledge engineering. Several methods including Bayesian networks, neural nets, genetic algorithms and truth maintenance systems exist but we are unable to cover all them in the scope of this paper.

So far we have presented our views on sensors and sensor systems, next we describe two additional components required to integrate sensor information in pervasive environments: location and situation awareness.

UNDERSTANDING LOCATION

Once the system has consolidated sensor data, the unified sensor view can be registered with locations or a geographical view. Semantic relationships associated with locations can provide facts which can be manipulated by the computer. While knowing the position of users in a pervasive environment is important, the ability to comprehend and reason about users' positions is more important. For example, GPS and electronic compasses are capable of returning accurate position and orientation information in most outdoor settings. Although it is sufficient for most users interacting with paper maps and for programs relying on raw coordinate data, abstract concepts such as "near my desk" or "in my kitchen" require additional knowledge not captured in this raw coordinate data. In the first case, "near my desk" implies that a "near" relationship exists between objects. The second example describes a boundary relationship. Boundaries can be difficult to describe, especially if the boundary areas need to be defined in sensor units. For example, a kitchen in a residential building is a less distinct area. Coupled with sensor ambiguity and noise, it becomes difficult to define an exact binary threshold for "in my kitchen." In most cases, we can construct a belief system composed of a system's level of confidence in device positions. Combining a mechanism for converting raw coordinate data into locations and semantic location information provides the necessary infrastructure for location-based reasoning.

However, some localization techniques do not require positional data. With sensor "landmarks," it is possible to determine locations by sensing a landmark. A landmark is an artifact introduced into the environment which associates a location with a unique name, position and semantic description. By noting the presence of landmarks in the sensor field of view and an appropriate mapping function, a system can deduce its location in the world. Unlike coordinate positions, locations are finite and usually have semantics attached to them. Landmarks are most effective when combined with spatial relationship operators like "adjacent to" and "above". As indicated previously, relationships allow the system to reason about users' relationships with their environments. Landmarks are not useful, though, in areas without features. Between major landmarks there may be large areas of featureless space. While it is possible to navigate between landmarks, it is not as easy as in a continuous coordinate based system.

Position and location information give the system a starting point for reasoning. In the next section, we describe how situation awareness encodes knowledge about user-user, user-location and location-location interactions.

UNDERSTANDING SITUATION AWARENESS

Situation awareness is defined as understanding the state of the system. It is not a new concept. Graphical user interfaces (GUI) and instrumentation engineers are aware of it. A GUI interface must understand the current state of the system to render the system's state. Situation awareness, as applied to a system, implies that the system understands the context and interrelationship between users and devices in the system and that the system has a set of rules or operations which it can apply at any time to determine how to react to a situation. While situation awareness can be implemented using "ladder-logic," a simple control technique, situation awareness is most useful when the control is flexible and where relationships, rules and actions can be modified easily. As with sensor fusion, situation awareness encompasses large tracts in the research literature and our suggestions cover just enough to convey its flavor.

PROBLEMS INTRINSIC TO SENSOR SYSTEM DESIGN

While each piece of a sensor tracking system has its own set of problems, often times many of the difficulties associated with building a sensor-based system occur once the pieces are integrated together. Probably the most difficult part of building a tracking and situation awareness system is intrinsically tied to the human aspect of the problem: managing the user's expectation. Any time a user interacts with the system, the user has an expectation of what the computer is doing. If the computer performs actions without the user explicitly taking action, the user becomes confused. This can cause the user to lose confidence in the system. Some of these expectations can be managed by enforcing a consistent design over all parts of the system. In our approach, we use sensors to monitor the situation and model the user's knowledge framework. By creating evolving models of the user's situation and knowledge, it may be possible to predict problems the user may have with an interface.

As with most interactive systems, it is important to integrate the user's feedback into the design. This may take the form of personalization menus or end-user studies. It is important to design mechanisms into the system which allow the system to be changed dynamically (and yet still be consistent with the user's expectation.) It is also important to have a framework for allowing the user to extend the system functionality without resorting to programming. Sensors can be used to help provide this functionality by monitoring the user's attitude and disposition. Picard's work in affective computing [7] is an example of this type of sensing. Likewise, situation awareness can be used to spot repetition in a user's routines. Repetition might be flagged as an inefficiency in the interface. Combined with the right UI mechanism, the system can propose changes in the interface design to the user.

CURRENT STATUS AND FUTURE WORK

Based on these observations, we have developed an initial prototype sensor system in support of pervasive computing. Our system consists of multiple active badges, landmark-based locations, a tracking and situation awareness system written in JESS [3], and an application layer written on top of the IBM Aglets server[6]. We are currently refining our sensor integration infrastructure incorporating many of the ideas discussed in this paper.

In the near future, we plan to add, evaluate, and integrate new sensors and tracking technology (i.e. biometric sensors and existing infrastructure like BlueTooth and pico-cell networking) into our sensor framework. We plan on integrating our sensor framework with existing pervasive computing efforts in intelligent messaging and ubiquitous display. Our overall goal is to produce a robust, portable system framework for supporting sensor-based systems, a reference guide containing detailed sensor evaluations and a system to support the design and building of sensor-based systems.

REFERENCES

1. Aboud, G. D., Atkeson, C. G., Brotherton, J., Engvist, T., Gully, P., and LeMon, J., "Investigating the capture, integration and access problem of ubiquitous computing in an educational setting," In Proceedings of the 1998 conference on Human Factors in Computing Systems – CHI '98, p. 440-447, May 1998.
2. Beadle, H. W. P., Maguire Jr., G. Q., Smith, M. T., "Using Location and Environmental Awareness in Mobile Communications," Proc. Of EEE/IEEE ICICS '97, Singapore, September 1997, pp. 1781-1785.
3. http://herzberg.ca.sandia.gov/jess.
4. Harter, A., and Hopper, A., A Distributed Location System for the Active Office, IEEE Network, pages 62-70, January/February 1994.
5. Kreller, B., Park, A. S., Meggers, J., Forsgren, G., Kovacs, E., and Rosinus, M., "UMTS: A Middle ware and Mobile-API Approach," IEEE Personal Communications, Vol. , No, 2, April 1998.
6. Lange, D. B., and Oshima, M., Programming and Deploying Java Mobility Agents with Aglets," Addison Wesley Longman Inc., Reading , MA, 1998.
7. Picard, R., Affective Computing, MIT Press, Cambridge, MA, 1997.
8. Voelker, G. M., and Bershad, B. N., "Mobisaic: An Information System for a Mobile Wireless Computing Environment," In Proceedings of the Workshop on Mobile Computing Systems and Applications, pp. 185-190, December, 1994.
9. Want, R., Schilit, B., Adams, N., Gold, R., Pertersen, K., Ellis, J., Goldberg, D., and Weiser M., "The PARCTAB ubiquitous computing experiment," CSL-95-1, Xerox PARC, March 1995.