Towards Situated Computing

Richard Hull
Hewlett-Packard Laboratories
Filton Road,
Stoke Gifford,
Bristol, United Kingdom
+44 117 979 9910
rh@hplb.hpl.hp.com

Philip Neaves
Hewlett-Packard Laboratories
Filton Road,
Stoke Gifford,
Bristol, United Kingdom
+44 117 979 9910
pin@hplb.hpl.hp.com

James Bedford-Roberts
Hewlett-Packard Laboratories
Filton Road,
Stoke Gifford,
Bristol, United Kingdom
+44 117 979 9910
jbr@hplb.hpl.hp.com

ABSTRACT

Situated computing concerns the ability of computing devices to detect, interpret and respond to aspects of the user's local environment. In this paper, we use our recent prototyping experience to identify a number of challenging issues that must be resolved in building wearable computers that support situated applications. The paper is organized into three areas: Sensing the local environment, interpreting sensor data, and realizing the value of situated applications. We conclude that while it is feasible to develop interesting prototypes, there remain many difficulties to overcome before robust systems may be widely deployed.

Keywords
Situated computing, situation aware, context aware, wearable computers, active tags

INTRODUCTION

Situated computing concerns the ability of computing devices to detect, interpret and respond to aspects of the user's local environment (see Figure 1). This capability promises both to add value to existing uses of computers and to create new types of application. For example, imagine a handheld tourist guide that can use its knowledge of the user's current location to present information about local sights, restaurants or other places of interest [1]. Or an eyeglass display that overlays peoples' faces with their names [2].

Situation awareness is particularly valuable for wearable devices. Desktop computers live in a very static environment. Even notebooks mostly only make the trek from the office to home and back. However, wearable computers will (potentially) go everywhere with their owners, into a wide variety of situations in which appropriate behavior for a given situation might be essential. Who really wants their mobile communicator to ring while in the midst of a theatre audience?

Indeed, we would argue that situated computing is an essential technology if wearable computers are to break the mould of today's notebook and palmtop computers. The potential for wearable computers is to be perceived not only as a physical extension of the user, but also as a mental extension. The difference between the two is whether the wearable computer is able to share our awareness of our surroundings and thus operate within a shared context, or whether the device remains "a stranger in the dark", albeit one that you are taking everywhere.

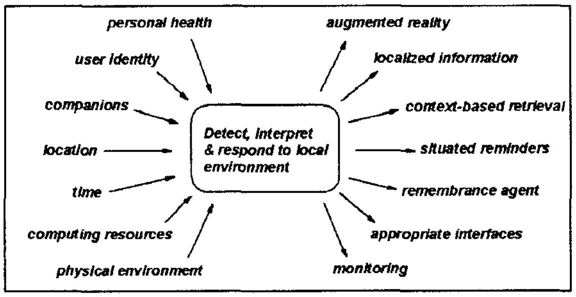


Figure 1 Overview of Situated Computing

Relationship to other work

Our main aim in this paper is to map out the wide range of issues that will influence the deployment of commercial situated computing systems. To uncover these issues we participated in the design and development of an experimental Ultra-Portable Computing[1] (UPC) platform, keeping a look out for general design issues as they arose. This gives our work a different slant to that of Xerox's EuroPARC [3] or "Ubiquitous Computing" initiative [4] where they develop bespoke systems as a means to investigating the potential of situated computing, rather than focusing on the systems themselves. In this respect

[1] a.k.a. wearable computing

our aims are similar to those expressed by Georgia's Cyberguide team [1] except that we focus more on architecture and infrastructure and consequently arrive at a different set of issues. We readily admit that we have not resolved all of the issues identified in our own work, but we hope that the paper might help others to identify important research topics.

In the course of this paper we describe an active tagging technology, a situated computing "server" and a "situated reminder" application. UPC systems that employ tags are not new (see the PARCTab system [5]). Our approach uses near-field radio to provide an alternative solution with unique properties (see below). The situated computing server brings out a set of services to interpret sensor data into useful events. This functionality needs to be present in all situated computing systems, but is not usually identified so explicitly. Our focus on the server reflects our emphasis on architectural issues. The situated reminder application was inspired by a strand of internal work at Hewlett-Packard Laboratories relating to "context-based retrieval" and "opportunistic task support". The same idea has also been implemented as a "Contextual Reminders" application on Xerox PARCTabs [6].

Structure

As indicated, the paper is divided into three main sections corresponding to the three main components of our current prototype. The first section discusses the use of an active tagging technology to sense places and people. The next section examines the problems in interpreting sensor data to provide applications with a stable, high-level view of the user's current situation. Then follows a discussion of how the value we hope to achieve from situated applications might be realized in practice. Finally, we attempt to draw some initial conclusions from our experiences.

SENSING PLACES AND PEOPLE

There are many aspects of a user's current situation that we might attempt to sense, for example, user identity, location, companions, vital signs, air quality, and network availability. For this work, we have concentrated on detecting two high-value dimensions - location and companions.

There are also many candidate sensing technologies available. Ideally, we would like a non-intrusive technology that did not require new infrastructure or cooperative colleagues. We are investigating recognition technologies that exploit some intrinsic characteristic of a person or place (such as speaker identification), but for our experiments we chose to use a simpler approach in which people and places are marked with a unique and clearly detected tag.

Marker tags can be passive or active. A passive tag does not require a power supply and consequently has advantages in terms of size, cost and lifetime. However, the range of passive tags is often limited to 1-2m and the tag detector is bulky and has high power consumption. For example, see the TIRIS system from Texas Instruments [6]. We require a small, low-powered detector that will fit into a wearable computer, and so must choose an active tag system. An active tag must use some wireless technology to transmit its identity to a detector - usually conventional radio or infrared [8][9]. In contrast, we have explored the use of near-field radio [10], a form of magnetic coupling that offers omnidirectional sensing and avoids reflection problems.

Design issues for active tagging systems

Whichever wireless technology is chosen, the design of the overall system must balance a number of considerations:

Size

The acceptable size of a tag depends on the thing it is marking. For example, a tag marking a meeting room could be fairly large, a tag carried by an individual should be no bigger than a conventional identity badge, and a tag marking a book might need to be as small as a postage stamp. A tag detector carried as part of a wearable computer also needs to be small, light and (perhaps) discrete.

Cost

To be useful, tags may need to be pervasive, either as a result of a coordinated deployment or simply of independent marking by individuals. Too high a cost will discourage widespread deployment. A unit cost of around $5 might be a reasonable target for a first generation product.

Power consumption

Although some tags may be mains-powered, most will rely on batteries. As we might expect a tag to be essentially free of maintenance, it might not be unreasonable to require a battery life of a year or so. The detector in a wearable computer will be similarly constrained, and will also have to share the available battery power with other system components.

Range

Typical applications for tagging include detecting proximity to some object (e.g. one's desk), or the presence of a tagged individual. In such cases, a range of a few metres seems appropriate.

Transfer capacity

The basic requirement is for a tag to be able to transmit its identity to local detectors. This might require only a few bytes of information to be broadcast. However, the tag might also be expected to transmit security information, a log of recent measurements, an individual's information profile, and so on. Moreover, it might need to transmit this data in a very quick burst to avoid clashes with other tags, increasing bandwidth requirements.

Error rate

Ideally, tags should be accurately and always detected. In practice, interference from background noise and other tags will corrupt and mask some transmissions. Although the system's interpretation software and applications should be able to cope with occasional errors, an overall error rate of around 1% seems a reasonable target.

Each of these design issues raises potential problems for the designer of an active tagging system. However, the real difficulty for the designer stems from the interdependence of the issues. For example, the bandwidth, and hence transfer capacity, of a tag is determined in part by the carrier frequency of the tag's transmitter. Increasing the carrier frequency would allow the tag's data to be transmitted more quickly, but would also increase the transmitter's power consumption and (perhaps) its component cost. On the other hand, a shorter transmit time would itself tend to reduce overall power consumption, and would reduce the number of errors that arise from clashes with other tags. In certain conditions, the size of the tag's antenna, and thus of the tag itself, would also be influenced by the choice of carrier frequency.

The choice of carrier frequency is just one of many decisions that the designer of the tagging system must make in order to optimize its performance with respect to the issues listed above. In the next section, we briefly review the outcome of the design process for our own prototype.

A prototype near-field tagging system: The Pinger

The Pinger active tagging system is made up of two components, a tag and a detector, coupled by a near-field radio link. Conventional radio systems operate in the far field of an antenna. However, all antennae have a near-field region. The precise definition of the near-field is somewhat complex and is a function of both frequency and antennae geometry. A common definition uses the distance r, $0 < r < \lambda/2\pi$, from the transmitting antenna [10]. Within the near-field region, energy transfer between a transmitter and receiver is via the electric or magnetic field component. A transmitter/receiver system using *loop antennae* will utilize the magnetic field component in much the same way as the coupling between the primary and secondary of a transformer, so allowing information (e.g. an identifier) to be passed from the tag to a detector.

The main advantage of near-field coupling stems from the restriction of operating range to a fraction of the carrier wavelength (6m at 8MHz). This arrangement virtually eliminates the problems of reflections and standing waves that disrupt tagging systems based on infrared or conventional radio.

Details of our approach can be found in [11], but we include some further discussion here to illustrate the general design issues made in the last section. The need for careful design is immediately apparent as the size, power consumption, and effective range of a near-field tagging system are intimately related. For an antenna of area A (m^2) in free space with a current of I (Amps) at a frequency of f (Hz), the maximum electric-field strength at distance r (m) is [12]:

$$E = 1.32 \times 10^{-14} \frac{IAf^2}{r} \quad \mathrm{Vm}^{-1} \qquad (1)$$

As may be seen, the electric-field strength is proportional to the antenna size, current through the loop, and the reciprocal of the distance to an interrogator. As the effective range is increased, either the antenna size must be increased to compensate, or the current (and power consumption) must be increased.

The design trade-offs are further complicated by regulatory requirements. A near-field tag is an intentional radiator and must comply with appropriate wireless telegraphy legislation. Table 1 shows the maximum current permitted through a loop of size 6 x 9 cm (approximately the size of an Hewlett-Packard identity badge) in order to comply with FCC part 15.223[2].

F (MHz)	$\lambda/2\pi$ (m)	I (Amps)
5	9.6	0.25
8	*6*	*0.1*
10	4.8	0.063

Table 1 Maximum allowed current for loop area of 6cm x 9cm.

The data packet broadcast by a tag consists of an 8-bit identifier, status and error correction bits, and is encoded using Pulse Width Modulation (PWM). The data packet is broadcast approximately once every second after which the tag enters a low power sleep mode to extend battery life. The characteristics of the prototype system are summarized in Table 2.

Frequency	8 MHz
Modulation	NBFM
Data rate	5 Kbps
Data format	8 bit tag value
Duty cycle	~1 Hz
Current Consumption	1 mA
Range	~ 3 m
Power supply	2 x AA batteries
Operational life	~ 70 days
Cost	$20 (300 up)
Size	6 x 9 x 1 cm

[2] In fact, the loop currents in Table 1 are fairly high, and might suggest that near-field radio is an unsuitable candidate for a low-power tagging device. However, if the loop is made resonant at the required frequency, the loop circulating current will now be a factor of the Q of the loop. Clearly, a single turn air-cored loop will have a low Q. Nevertheless, even a moderate Q (e.g. 10) will significantly reduce the power consumption of a near-field tag. Power consumption can be reduced further by causing the tag to only transmit its identity once every second or so.

Weight	100 g

Table 2 Characteristics of the prototype tagging system

This configuration provided a prototype tagging system in which around 95% of pings emitted by a tag were successfully detected at a range of 3m. The most important external factor influencing this performance is the presence of local Electro-Magnetic Interference (EMI). Offices contain multiple sources of EMI such as monitors, notebook PCs and electronic ballasts in fluorescent lighting. A full characterization of the system under typical office conditions may be found in [11].

The prototype system works well though continued development is still needed to satisfy all of the design requirements outlined earlier. For example, the system is more expensive and consumes more power than we would like. To emphasize our earlier point, this reflects the need to make difficult design trade-offs in a multi-dimensional design space. Moving to a different technology, such as infrared, would not suddenly ease the design constraints but simply lead to a different set of trade-offs.

Tag deployment

Underlying the last section has been the assumption that enough tags will be deployed in the environment to make a detector a worthwhile investment for the wearable computer owner. This begs the questions of how the tags are deployed, and whether any infrastructure is required to support their deployment. Who will pay for the tags in the first place? Or ensure unique identifiers? We will return to these and related issues later in this paper.

INTERPRETING SENSOR DATA

Sensors such as the active tagging system just described provide raw data about the local environment. The information inherent in this data will eventually influence the behavior of situated applications sensitive to that environment. However, the data typically provided by sensors will not be in a form that can be used easily by applications. For example, a sensor may report that a particular tag has (probably) been detected whereas an application really wants to know that the user has just been approached by one of her colleagues. To become useful to applications, sensor data must be interpreted to provide a stable, reliable and abstract view of the status of the current situation, and changes in that status. In a UPC operating system, this is the function of the Situated Computing (SitComp) Service - a piece of middleware utilizing local sensors to provide situational information to applications via a standard API. This organization is shown in Figure 2.

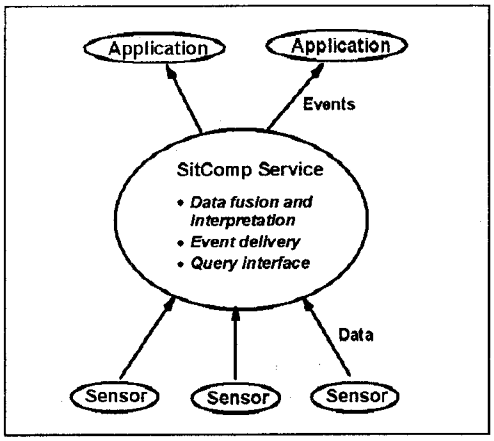


Figure 2 Overview of Situated Computing Service

The main responsibilities of the Situated Computing service are:

- To combine and interpret incoming data from sensors so as to provide situational information at an appropriate level of abstraction for situated applications.
- To post events to interested applications whenever some aspect of the current situation changes.
- To provide a query interface to allow applications to interrogate the current situation.

Underlying this statement of responsibilities is the event-driven model of computation familiar from systems providing a graphical user interface (GUI). In a GUI system, an operating system service monitors the interface and sends events to an appropriate application whenever a user action (such as clicking a button) is detected. Similarly, in a situated computing system, the SitComp service monitors the sensor data and sends events to appropriate applications whenever a significant change is detected in the local environment. Of course, the SitComp service (and indeed an interface service) must also provide applications with the ability to query situational information independent of particular events.

Given this overall model, there are a number of system requirements and interpretation issues to be faced.

System requirements

In a UPC system supporting a number of applications using situational information derived from multiple sensors, the SitComp service may need to satisfy (at least) the following requirements.

Transparency

Situational information derives from sensor data but the sensors available to a UPC (or any other wearable computer) are likely to vary as users select different system components to carry according to circumstances. An operating system typically hides such configuration details from applications through abstract services. The SitComp service should follow this model by providing applications with transparent access to situational information when any capable sensor is available. For example, a situated reminder application can be informed about the identity of a user's companion, regardless of whether that companion happened (this time) to be detected through an active tag, speaker identification, or face recognition[3].

Dimensional independence
The SitComp service will typically be expected to provide information on many aspects of a user's current situation. Each situational dimension should be managed independently wherever possible. In particular, problems or delays in providing information on one dimension should not affect the provision of information about another. For example, a situated reminder application based on companion information should not be blocked while a lengthy determination of the user's precise location is determined for a route finding application

Low latency
The value of many situated applications will depend on a rapid response to changes in the local environment. This may be obvious for an application monitoring, say, toxic emissions, but it could also be true for less critical applications like situated reminders. If a UPC is going to remind its user to pass on a message to a companion, then it better do so before the companion has moved away. Hence, the SitComp service should signal changes in the current situation soon after they come into effect. This means giving high priority to the processing of situational events[4], but also making a careful trade-off between response time and accuracy for event detectors, such as speaker identification, that work best when allowed a few seconds (or minutes) of data.

High throughput
Not only must a SitComp service respond quickly to a particular situational event but it must also be able to handle many concurrent events. For example, imagine being in an audience at a big presentation in which everyone has an active tag transmitting their identity. The SitComp service might have to handle hundreds of pings a second, identifying their senders, deciding whether any of the pings indicate a change in the situation, and posting events as appropriate. Bear in mind too, that this might have to be a background activity.

Extensibility
The UPC is intended to be a general computing platform. We cannot predict or constrain exactly what it would be used for and in what configuration. Even if the base system was shipped with in-built sensors that allowed it to provide, say, location information, the owner is likely to want to extend its capabilities. This might mean wanting to install a third party package providing an entirely new situational dimension (e.g. heart rate), or it might mean installing a new class of sensor for an existing dimension (e.g. adding speaker identification to a system already using active tags for companion detection).

Interpretation issues

The purpose of interpretation is to provide applications with a stable and up-to-date view of the user's current situation, at an appropriate level of abstraction, based on the data coming from a variety of sensors. This objective is straightforward but very challenging for all except the simplest cases. Consider just a few of the difficulties that must be overcome:

Noisy data
The data provided by sensors is unlikely to be either completely available or completely correct. For example, pings from an active tag may be missed, or corrupted by background noise. Data on a continuous scale, like temperature, will only be accurate within some error tolerance. Data from recognition-based detectors, such as speaker identification, will be probabilistic[5], and often wrong.

Redundancy
Any particular UPC configuration may have a number of different sensors capable of signaling some aspect of the local environment. For example, the identity of a companion might be derived from active tags, speaker identification, or face recognition. Diversity of sensing is an advantage - for example, speaker identification might detect a companion who has forgotten to wear their tag - but it does require some method of combining evidence from an (unpredictable) set of sensors with different characteristics. Well-known data fusion techniques exist for the case of multiple sensors of a single type but the problem of combining evidence from diverse sensors at different levels of abstraction is much less well understood [13].

Brittle heuristics
The examples of situated inferences related in this paper imply fairly simple heuristics. For example, a voice is heard and identified, and the speaker is assumed to be present. But there other possible explanations: Perhaps the voice is from a recording, or someone shouting far away, or a friend talking to someone nearby (i.e. present in body but not in spirit), or an impersonator... Simple heuristics are liable to produce wrong inferences in many situations. However, the alternative - building interpreters capable of commonsense reasoning about the world - remains infeasible.

These three issues alone are sufficient to alert us to the challenges in interpreting data sensed from the local environment. Of course, such problems are tackled

[3] Of course, in exceptional circumstances an application may wish to know how a piece of situational information was derived and should be able to find out. However, our general point stands.

[4] And, perhaps, the use of an interrupt model to signal high priority events to applications?

[5] Or more correctly, will be associated with a confidence level, which might be treated as an indication of probability of correctness.

routinely by the developers of measurement systems. However, the UPC context is made much more difficult by the intended generality and dynamic configurability of the target devices. Unlike a bespoke measurement system, neither the exact set of sensors available at runtime, nor the precise application of situational information will be known to the developer of interpretation software embedded in the SitComp service.

An experimental SitComp service

We have explored some of these issues by prototyping an experimental SitComp service with an emphasis on transparency, dimensional independence, and extensibility. Its principal function is to maintain a dynamic network of connections between sensors, interpreters, and situated applications. As sensor data flows up through this network, it is combined and abstracted by interpretation layers. Eventually, the dataflow impacts a level of abstraction exposed via the service's API and an appropriate event is posted to interested applications.

All entities in the network join by registering with the SitComp server as sources or consumers of situational information. Interpreters are simply application level processes that consume one type of information (e.g. pings) and source another (e.g. companion events). New sensors, applications and interpreters may be added (or removed) at any time, providing extensibility and robustness against configuration changes. Applications that have registered an interest in a particular situational dimension receive all events for that dimension regardless (and without knowledge) of their original source.

To date, we have placed less emphasis on developing sophisticated interpreters beyond acknowledging the issues outlined earlier. Data fusion and abstraction is achieved by enforcing standard formats for situational dimensions, and ensuring that the output of all entities sourcing a dimension are routed to an appropriate interpreter. Thus, for example, an interpreter monitoring the presence of companions could receive *PersonSpotted* events that originate from both an active tag detector and a face recognizer. However, we have only utilized a single sensor type (active tag detectors) so far, and cannot claim to have seriously addressed the issue of combining information from such disparate sources.

REALIZING THE VALUE OF SITUATED APPLICATIONS

Situated computing is not a single value or even a single application. Like "image processing" or "pattern recognition", it is a software technology that can be applied to a wide range of application areas, including:

- *Augmented reality*. For example, a user may be able to see how a flowering cherry tree might look in her garden by using a head mounted display and a situated application that adds the tree at a predefined location and scale. Note that the application needs to monitor the user's situation so that it can render the tree in the correct place within the display area.
- *Localized information* allows a situated computing device to answer questions such as "Where is the nearest pizza restaurant?"
- *Context based retrieval* allows people to retrieve information according to an associated physical context. For example, a user might request "Get me the document I was reading on the train yesterday".
- *Situated reminders*. For example, "Remind me to buy some toothpaste when I am next near a supermarket"
- *Appropriate behavior* of devices. For example, a voicemail application that pauses audio playback when a colleague enters your office, or a personal organizer device that disables audio alarms while in a meeting room or lecture hall.
- *Monitoring*. For example, a medical device that monitors the heart rate or body temperature of a hospital patient.

Measuring value is hard

We would like to understand which applications of situated computing offer the highest value to end users. However, this is not a straightforward issue.

First, identifying possible application areas is not the same as understanding the potential user value in each area. This is partly because the areas are still too broad to assign a single value to each, and partly because many applications are tool-like in that their value depends on how creatively they are applied. In order to determine the potential value of a tool-like application, you need to hand out the application to a large number of trial users.

In addition, social issues strongly affect the adoption of the pervasive computing platforms that are needed to support situated computing applications. For example, people are concerned about their public image. This may make them feel uncomfortable interacting with novel devices in public places. People also value their privacy. This may make the uncomfortable having their situation monitored by electronics that they may not fully understand or control. Both these sources of discomfort represent costs that undermine the value of situated computing applications. Users can hold widely differing, wildly conflicting and strongly held views in this area [14].

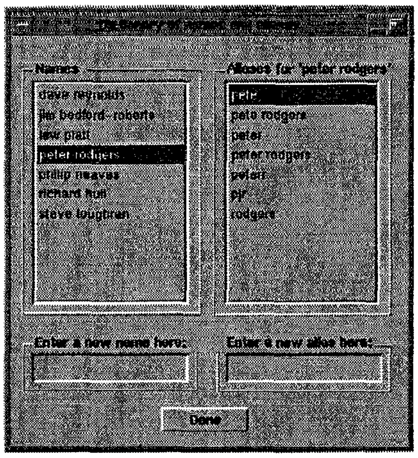
Lastly, situated computing may not always address the main benefit of an application directly, but may add value either by adding serendipity or by reducing costs. For example, consider the MIT remembrance agent [15]. This agent operates inside a note-taking application from where it monitors the themes the user is currently entering and offers links back to previous notes with similar themes. It is possible to imagine a future, situation-aware remembrance agent, embedded in a multi-media capture device, that is able to cross reference different media objects captured in related situations or with related themes. The situatedness of the application gives the opportunity to offer serendipitous value while the user is primarily engaged in note taking. On a more mundane level, a voicemail application that goes quiet when a colleague enters the room adds extra value through decreased cost of use rather than by increased user benefit. In all these cases, it may be hard to assess the contribution of situated computing to the overall value of the main application.

Infrastructure may be required

Situated computing applications may have large initial costs. Even a quite modest application might require distributed information infrastructures, or networks of sensors and transmitters, if its use is to become widespread.

Information infrastructure

Situated computing applications often require the end-user to specify a particular situation. For example, users may be required to describe the situation in which a situated reminder is to be triggered or the class of situations where audible alarms are to be disabled. Language is the usual way to specify abstract concepts such as situations, but is open to ambiguity. Even for a simple application that raised reminders in the presence of specified companions, we needed to provide a way of disambiguating multiple aliases for people into unique names (see Figure 3).



dave reynolds
jim bedford-roberts
lew pratt
peter rodgers
philip neaves
richard hull
steve loughran

pete
pete rodgers
peter
peter rodgers
peterr
pjr
rodgers

Figure 3 Interface for a prototype alias server

Aliases and "unique" names are always associated with domains in which they are valid. Any naming scheme that ignores domains is doomed to eventual scaling problems (running out of names). Establishing a naming domain hierarchy requires a range of management and standardization issues to be solved. Thus even quite modest situated computing applications might need to "pay up front" if they are to avoid scaling problems. Even if the full infrastructure does not need to be established initially, it is certainly necessary to have solved the design problems that will enable it to be built in due course.

Physical infrastructure

A physical infrastructure can provide networks of devices (e.g. transmitters, tags, sensors, satellites and servers) that may be shared by many different applications to provide diverse values to end users. In the long term, this reduces the marginal cost of the infrastructure for any one application to acceptable levels. However, the initial cost of installation must still be borne by a particular usage before any other applications can be deployed. For an infrastructure of national (or even corporate) scale, this initial investment might be prohibitive.

Deployment dynamics

These infrastructure issues lead to some interesting possible dynamics in the way that situated computing applications could spread. The early applications could include:

- *Situated applications that do not require device infrastructure.* For example, applications that recognize people using face recognition technology (though naming issues remain).
- *Situated applications where the infrastructure already exists.* For example, applications that use Global Positioning Satellite systems to provide location information.
- *Situated applications where value can be obtained even when device networks are just local.* For example, some businesses may be able to justify investing in active badges for all staff on a given site if particular productivity improvements will justify the cost.
- *Situated applications in vertical markets.* For example, a chain of restaurants or stores may be prepared to subsidize a device capable of providing customers with directions to their nearest outlet.

A second wave of applications could then emerge based on the infrastructure associated with these first applications. There may be an additional surge as ubiquitous computing devices become more common and their "social profile" shrinks. All this could lead to a spiral of uptake that may be quite different from expectations informed solely by the value to end users that different applications could provide.

CONCLUSIONS

In this paper we have attempted to raise a number of issues associated with the development and deployment of situated computers. These might be summarized as follows:

- A sensing technology capable of detecting places and people has to satisfy a number of interdependent design requirements such as size, cost, range, and power consumption. Active tagging technologies offer one promising approach and our experiments suggest that near-field radio might be a basis for such a system.
- Interpreting sensor data is vital but hard. Situated applications need a stable view of the user's local situation (and changes in that situation), at an appropriate level of abstraction. Sensor data is often noisy, unreliable and redundant, and describes the local environment at too low a level of detail. Bridging the gap is a difficult reasoning problem that seems to

require a level of commonsense beyond simple heuristic systems. The problems of interpretation are exacerbated by the need for a general-purpose wearable computing platform to support a dynamic configuration and an unpredictable set of applications.

- Realizing the value of situated applications will depend on a better understanding of the social issues involved and a deployment strategy that balances the need for infrastructural investment against the cumulative benefits added by making computers situation-aware. Given this context, we identify a number of ways in which situated applications might first be deployed.

We have discussed the issues in the paper in the context of our own experiences in prototyping a general-purpose platform for Ultra Portable Computing that supports situation-aware applications. The requirement for generality is what makes situated computing difficult compared to a bespoke sensing system, but it is the necessary consequence of a capability that might most often add value to existing uses of computers rather than define wholly new applications.

Our current view is that while it is feasible to demonstrate interesting situated computing prototypes, there are many difficulties to overcome before we are able to build robust, valuable systems for widespread deployment. We hope that by raising these issues, we might encourage others to tackle some of those difficulties and so fulfill the potential of situated computing. In the meantime, we are developing a "Bristol Tourist Jacket" in collaboration with Bristol University and plan to use this as a platform for further work in situated computing.

REFERENCES

1. S. Long, D. Aust, G. Abowd and C. Atkenson, Cyberguide: Prototyping Context-Aware Mobile Applications", *Proc. CHI 96*, ACM, April 1996, pp. 293-294.

2. T. Starner, S. Mann, B. Rhodes, J. Levine, J. Healey, D. Kirsch, R.W. Picard, and A. Pentland. "Augmented Reality Through Wearable Computing*", To appear in Presence, Special Issue on Augmented Reality.*

3. M.G. Lamming, W.M. Newman, "Activity-based Information Retrieval: Technology in Support of Personal Memory", *Personal Computers and Intelligent Systems. Vol. A14 of IFIP 12th world congress. Proceedings of Information Processing '92*, pp68-81. IFIP, Elsevier Science Publishers (Holland) 1992.

4. Weiser, M. "Some Computer Science Issues in Ubiquitous Computing", *Communications of the ACM*, July 1993.

5. R. Gold, M.M. Tso, R. Want, "The PARCTab Mobile Computing System", Xerox PARC, 1993.

6. B. Schilit, N. Adams, R. Want, "Context-Aware Computing Applications", *Proc. Mobile Computing Systems and Applications*, IEEE Comput. Soc. Press, 1995, pp. 85-90.

7. U. Kaiser, W. Steinhagen, "A Low-Power Transponder IC for High-Performance Identification Systems", *IEEE Journal of Solid-State Circuits*, vol. 30, no. 3, 1995, pp. 306-310.

8. A. Harter and A. Hopper, "A Distributed Location System for the Active Office", *IEEE Network*, Vol. 8, No. 1, Jan/Feb 1994.

9. R. Want, A. Hopper. "The Active Badge Location System", *ACM Transactions on Information Systems*, Jan 1992, pp. 91-102.

10. A. Demers, S. Elrod, C. Kantarjiev, and E. Richley, "A Nano-Cellular Local Area Network Using Near-Field RF Coupling", in *Wireless Personal Communications*, ed. B.D. Woerner, T.S. Rappaport, and J.H. Reed, Kluwer, Norwell MA. USA, 1995.

11. P. Neaves, "An Active Tagging System Based on Near-Field RF Coupling", *In preparation.*

12. P. Lorain, D.R. Corson, and F. Lorrain, "*Electromagnetic Fields and Waves*", W. H. Freeman and Company, New York, 1988.

13. M. Kam, X.X. Zhu and P. Kalata, "Sensor Fusion for Mobile Robot Navigation", *Proc. Of the IEEE*, Vol.85, N1, 1997, pp. 108-119.

14. R. Harper, "Why People Do and Don't Wear Active Badges", *Proc. CSCW '96*, Kluwer Academic Publisher, pp. 297-318.

15. B.J. Rhodes, T. Starner, "Remembrance Agent: A continuously running information retrieval system" *Proc. First Int. Conf. on The Practical Application of Intelligent Agents and Multi Agent Technology (PAAM '96)*, pp. 487-495.