

# Adding Generic Contextual Capabilities to Wearable Computers

Jason Pascoe

Computer Laboratory, University of Kent at Canterbury,  
Canterbury, Kent CT2 7NF, United Kingdom

+44 (0)1227 764000 ext.7754

J.Pascoe@ukc.ac.uk

## Abstract

*Context-awareness has an increasingly important role to play in the development of wearable computing systems. In order to better define this role we have identified four generic contextual capabilities: sensing, adaptation, resource discovery, and augmentation. A prototype application has been constructed to explore how some of these capabilities could be deployed in a wearable system designed to aid an ecologist's observations of giraffe in a Kenyan game reserve. However, despite the benefits of context-awareness demonstrated in this prototype, widespread innovation of these capabilities is currently stifled by the difficulty in obtaining the contextual data. To remedy this situation the Contextual Information Service (CIS) is introduced. Installed on the user's wearable computer, the CIS provides a common point of access for clients to obtain, manipulate and model contextual information independently of the underlying plethora of data formats and sensor interface mechanisms.*

## 1. Introduction

Context-awareness [1] is the ability of a program or device to sense various states of its environment and itself. In one form or another, context-awareness is already a basic requirement of many systems and is likely to have an increasingly important role to play in future wearable computers and in the integration of wearable and ubiquitous computing. The intimate association of user and computer in a wearable system leads to the computing resources being accessed in a diverse array of situations, unlike a static desk-bound computer. It is this multitude of dynamic contextual factors that allows context-awareness to be exploited particularly well in wearable computers.

This paper addresses the deployment of context-awareness in wearable computers through three sections. The first identifies four application-independent contextual capabilities that can be employed to enhance the wearable system. In the second section a case study is given that demonstrates how even a relatively simple context-awareness can often greatly assist the user. The final section introduces the Contextual Information Service, which, based on our development experiences, we believe is vital to facilitate the wider use of context-aware technology.

## 2. Relationship to other work

The case study shares a general theme of exploiting a knowledge of time and place that is in common with other applications such as the touring machine [2]. However, most applications currently use this knowledge to influence how information is *viewed* (e.g. displaying some tourist information) whereas our prototype application uses context-awareness to influence how data is *recorded*.

The design of a Contextual Information Service (CIS) is a largely overlooked area of research. Contextual services tend to be of an application-specific level, such as Cyberguide's navigator and cartographer [3]. Other projects that attempt a more application-independent approach, such as the situated computing service [4], tend to offer a service that simply abstracts the client application from the underlying sensors. Our CIS is designed to provide a more comprehensive service that models contexts in terms of the objects they belong to and the relationships between them. Rather than a conveyance medium operated by discrete calls to a generic sensor API, our CIS seeks to provide a continuous service via a rich contextual model that happens to gain some of its data automatically from sensors.

### 3. Generic context-aware capabilities

Schilit *et al* originally defined a set of general context-aware application categories [1], and subsequent research projects have defined application specific categories. However, what we present here are a set of core generic capabilities that we use as a vocabulary to identify and describe context-awareness independently of application, function, or interface.

#### 3.1 Contextual sensing

Contextual sensing is the most basic level of context-awareness. The wearable computer simply detects various environmental states and presents them to the user in a convenient form, effectively augmenting the user's own sensory system. For example, if connected to a location sensor such as a Global Positioning System (GPS) receiver [5], then the derived latitude and longitude location can be conveyed to the user via a map display annotated with a 'you-are-here' marker.

#### 3.2 Contextual adaptation

It is not only the user that may be interested in the contextual data from such a sensory system. Applications can leverage this contextual knowledge by adapting their behavior to integrate more seamlessly with the user's environment. Rather than providing a uniform service regardless of the user's circumstances, the context-aware computer can tailor itself to the current situation. For example, adapting behavior for a particular user (where the user's presence is the context), turning the screen back-light on when it gets dark, switching to a communications link that is less expensive during the current time of day, etc.

#### 3.3 Contextual resource discovery

Contextual adaptation applies a knowledge of the wearable computer's own context (which, for the most part, also inherently equates to the user's context). This is taken a step further if the contexts of other entities are made available. Using this information the wearable computer can discover other resources within the same context as itself and exploit these resources while they remain in the same context.

The resources that are most readily useful to the wearable computer are various forms of electronic or

computer equipment. For example, the user's wearable computer with limited display capabilities may discover an unused display screen that is in close proximity and can temporarily be used to display information on. Less conspicuous *information resources* may also lie within the user's environment. For example, a service timetable may be broadcast by a local infrared transmitter installed at a bus-stop. If the transportation service itself is context-aware [6], then real-time estimates of arrival times could be calculated based upon its present position.

An important aspect of contextual resource discovery is that it bridges the philosophical gap between wearable and ubiquitous computing [7]. The wearable computer provides a set of core services constantly at the user's disposal, whilst also contextually binding to devices in the user's environment (i.e. ubiquitous devices) that can complement or augment the wearable system.

#### 3.4 Contextual augmentation

Thus far, we have discussed how contextual information can be used to sense, react, and interact with the environment. Contextual augmentation extends these capabilities further through augmenting the environment with additional information. This is achieved by associating digital data with a particular context that it is related to. Depending on the user's perspective this coupling of the real and the virtual can be either viewed as the digital data augmenting reality [8] or reality augmenting the digital data.

Tour-guides are the most prevalent applications of augmenting reality with digital data [2, 3]. The tourist equipped with such an application is presented with information about the attractions that they are surrounded by or are approaching. Considering the alternative perspective of digital data augmented with reality, an address book program could augment an entry with the addressee's present context, such as their current location and activity. Given this extra information, the user can route a phone call to the phone nearest the addressee or postpone the call if the addressee is currently in a meeting.

In addition to reality-*information* couplings, there are also reality-*process* couplings. These facilitate the execution of a program or particular process, when the user happens upon a specific situation. In effect, this extends the event-driven architecture common in the majority of computer systems to encompass contextual events of the surrounding environment.

## 4. A giraffe observation case study

A prototype fieldwork tool is examined in this section, produced as part of the ‘Mobile Computing in Fieldwork Environments’ project [9, 10] at the University of Kent, funded by JTAP. This project has experimented with the aforementioned context-aware concepts by constructing prototype systems to assist users in common fieldwork tasks that are typical across a broad spectrum of subject domains (e.g. an archeological excavation, a biodiversity survey, etc.). The particular system examined here has been used extensively by an ecologist during a two-month observational study of giraffe in Kenya.

### 4.1 Why fieldwork?

Wearable and context-aware computing concepts are particularly appropriate in the domain of the fieldworker. The user may traverse large areas on foot, crawl through thick undergrowth, or face other inhospitable conditions that make carrying or operating a laptop computer completely impractical. Therefore, a small, lightweight, robust, device that can be used on the move is required, i.e. the wearable computer. Additionally, the extremely dynamic user-environment also equates to a diversity of fast changing contexts that can be exploited by the context-aware wearable computer. For example, one aspect of the giraffe observations was to record how many bites a giraffe ate, where it ate them, and at what time – information that changed from second to second.

The need for a new approach to the provision of computing resources is not solely driven by the unusual environment of the user, but also by the nature of the work itself. In most field observation work, the user needs to maximize the time they spend observing whilst minimizing the time they spend on auxiliary tasks such as interacting with the recording medium. This applies even more so when there are large volumes of data to be recorded within short periods. A context-aware wearable computer helps through its intimate association with the user, attempting to seamlessly integrate the computing resource with the user, their task, and their environment.

Finally, data collection and observational work (which comprise the larger part of fieldwork) focus on recording various states of the user’s environment, e.g. the location of an observation point. A context-aware recording device can thus greatly assist the user in making observations because it too can ‘observe’ some states of the user’s surroundings.

### 4.2 Field assistant hardware

Our directive for work on the first prototype was to explore and develop context-aware software rather than construct new hardware designed for the fieldworker’s ergonomics. Our solution reflects this attitude, comprising a selection of readily available hardware running the novel fieldwork tools that we have developed.



**Figure 1. The Ecologist records an observation of a giraffe (see top right of photograph) using the 3Com PalmPilot linked to GPS-enabled trousers.**

We restricted our selection of computing hardware to low-cost devices that were widely available and that best matched the earlier described attributes of being small, lightweight, and robust. We chose the 3Com PalmPilot PDA (Personal Digital Assistant) as its small notepad-like form factor, its Graffiti handwriting recognition system, and its design philosophy oriented towards simple, easy-to-use functionality, singled it out as the most suitable candidate for replacing the fieldworker’s paper notebook. Already aware of its time context, we augmented this with a knowledge of location by attaching a Garmin-45 GPS receiver. This set-up provided the foundation on which to build programs that utilized a simple understanding of time and place.

The ecologist’s army trousers were equipped with the PalmPilot and GPS receiver in the left and right trouser-leg pockets respectively, where they remained until the user removed the PalmPilot in order to note down an observation, see figure 1. An unobtrusive umbilical cable

kept the two stowed devices constantly connected. To improve the performance of the GPS receiver an external aerial, which was discretely concealed in the fabric of a baseball cap, could also be connected. However, in the giraffe study, the ecologist preferred to use the GPS receiver without this optional accessory.

#### 4.3 Field assistant software

The prototype software currently concentrates on providing a set of tools to assist in the fieldworker's observation and data collection activities, i.e. helping the user record information about their environment. This is an interesting divergence from other context-aware studies, such as the oft-quoted location-aware tourist guide, as here information is being *authored* in a particular context rather than *presented* in a particular context.

The *stick-e note* context-aware model [11] was used as the basis for constructing the fieldwork tools. In this model, electronic notes that are made by the user are automatically attached to the current context (typically the user's location) thereby creating a stick-e note. For example, a stick-e note could consist of a 'Saw giraffe here' message and the related location context the message is attached to. This model was extended and generalized so that a stick-e note now consists of a set of elements that describe both the content *and* context of the note. This is exactly what an observation is: a set of values that equally describe the content or context of the observation depending on one's perspective. For example, an observation note could consist of a time, location, and animal field, each of which describe both content *and* context. We developed several programs for the PalmPilot that allowed the user to create, edit, and view stick-e notes in the field, the two main ones are detailed here:

*The StickePad.* All the creating and editing of stick-e notes is handled by this program. The user can create a new stick-e note from a template (a set of name and type pairs created by another program) and then edit the values in a form-based user interface, see figure 2.

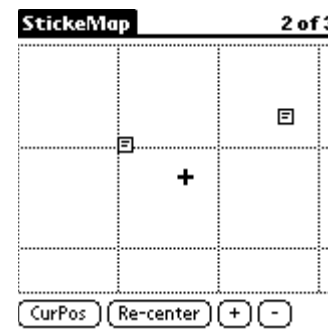
To assist in the observation process the StickePad employs the system's knowledge of time and place to automatically default any location or time fields to the current contextual values. The user can still edit these fields manually or request that they be updated with new contextual readings (useful if the user changes context whilst editing the note). We also deployed a number of

HCI techniques in this stick-e note authoring tool to optimize it for the fieldwork environment, utilizing principles such as indirect modes of operation [12]. However, only the context-aware features will be discussed within the scope of this paper.

The screenshot shows a form titled "Sticke Edit" with a page indicator "1 of 18". It contains several input fields: "Date:" with value "30/9/1997", "Time:" with value "08:01:26", "GPS:" with value "0 1.928 N 36 53.2", and "Direction:" with value "060". Below these is a sub-form titled "Edit Lat & Longitude" with a help icon. This sub-form has two rows of input fields for latitude and longitude, each with a north/south or east/west indicator. The first row shows "0 0" for degrees and "0 1 9 2 8" for minutes, with "N S" as the indicator. The second row shows "0 3 6" for degrees and "5 3 2 5 7" for minutes, with "E W" as the indicator. At the bottom of the sub-form are three buttons: "OK", "Cancel", and "Here".

**Figure 2. Manually editing a location field of a stick-e note in the StickePad. The 'Here' button retrieves the current GPS value.**

*The StickeMap.* A knowledge of the location context of both the user and the recorded observations is exploited in the StickeMap to visualize the collection of stick-e notes on a map display, see figure 3. There are no background details such as tracks, hills, etc. as no mapping data was available for the area at the time of constructing the prototype. However, future versions will incorporate mapping data that is currently being generated.



**Figure 3. The StickeMap shows the user's current position (the '+' icon) relative to the stick-e notes (the note icons).**

Selecting one of the stick-e note icons displayed in the StickeMap invokes the StickePad with the appropriate stick-e note automatically loaded for editing. This offers an alternative stick-e note selection mechanism to the simple sequential note list available in the StickePad.

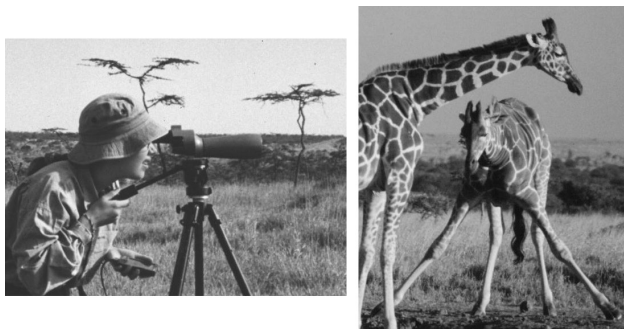
#### 4.4 The field trial

The ecologist used the prototype for all of her data collection requirements throughout the course of the two-month giraffe study. The work carried out during this period can be divided into three main observational activities, each associated with a particular stick-e note template:

*Giraffe Observation.* Studying groups of giraffe and individual giraffe behavior (for which two stick-e note templates were constructed by the ecologist) was by far the most common and intensive activity. It would normally have required the work of two people, one to make observations and the other to take notes. However, the automatic entry of location and time contexts into the stick-e note eliminated the need to operate a stop-watch or to estimate position (very difficult to do in a homogenous environment). Additionally, the computer and telescope could be operated simultaneously, see figure 4. Thus, the workload was reduced to such an extent that the single ecologist alone could complete the task.

*Vegetation Surveys.* The aim of these surveys was to establish the impact of giraffe feeding on the Acacia tree in different parts of the reserve. Recording the information in a stick-e note that was automatically tagged to location proved an effective solution.

*Fecal Sampling.* Samples of giraffe feces were routinely collected for analysis in order to investigate giraffe diet. Stick-e notes were used to record the origin of any samples by automatically providing the location context to accompany the description of the state of find.



**Figure 4. Recording giraffe behavior.**

In addition to automating elements of the conventional data collection process in each of these activities, visualizing the location context in the StickeMap provided novel functionality applicable across all of these tasks. The map display served two purposes:

(i) to allow the ecologist to orient and locate herself relative to the stick-e notes she had recorded, and (ii) to observe any patterns in the distribution of such notes, e.g. to see that the majority of feces had been collected in a particular area of the reserve.

The hardware configuration also proved a good design for the environment. The lightweight, small, and robust attributes of the equipment allowed it to be used effectively in the harsh conditions without physically impeding the user in any way. When making an observation the device could be operated with one hand, and at other times kept in a pocket. This ability to completely remove the device from the user's awareness was essential when walking through areas inhabited by rhino and lions. In such conditions the user could forget about the computer system and devote their whole attention on avoiding an undesirable confrontation with a rhino, or, failing that, running away from one as fast as possible! Occluding any of the user's senses or mobility, even partially, is simply not acceptable in this environment (though in less hazardous fieldwork settings this is not necessarily the case). In summary, it was desirable to have the computing resource as transparent as possible, effectively removing it when not required, and providing it in a convenient, easily accessible, manner when it was.

At the end of the trial approximately 6000 individual observations had been recorded using the prototype. The ecologist stated that the system had allowed her to record more information, quicker and easier than would otherwise have been possible, proving itself a successful solution to the fieldworker's needs.

#### 4.5 An evaluation of context-awareness

Although the prototype proved to be very useful, how much has context-awareness contributed to this success? In this first prototype, contextual-awareness is limited to two types of context: time and location, and it only exploits two of the four general context-aware capabilities described earlier: contextual sensing and contextual augmentation. However, these limited abilities were effectively targeted on the areas of automating data collection, transparently providing contextual information when required, and visualizing collections of observations whilst in the field. Deploying context-awareness in these areas was a key factor in greatly improving the speed and volumes of data collection, and in providing new opportunities for the fieldworker, such as assessing the

distribution of observations whilst still in the field. In short, the context-aware capabilities were central to the system's success.

In the next generation of prototypes we would like to enhance and add to the level of context-awareness. A few suggestions are presented here, categorized by the type of capability.

*Contextual Sensing.* The most obvious improvement would be to add more types of contexts. Remote contexts could also be employed given a wireless communications link, e.g. querying a remote GIS or meteorological station to obtain contextual data about the user's current location.

*Contextual Adaptation.* Not used at all in the current prototype, this capability could provide the fieldworker with assistance by automating actions in certain contexts. For example, the user could prepare a data collection strategy that determined the contexts for sample points in advance. The fieldwork tool could then automatically invoke a new stick-e note whenever the user enters the appropriate location, time, or other specified context.

*Contextual Resource Discovery.* This is of limited use in the Kenyan fieldwork setting due to the general lack of any man-made infrastructure. However, although devoid of physical *computing* resources to discover, there are possibilities in discovering *information* resources. For example, a biography of a rhino discovered via the animal's radio collar.

*Contextual Augmentation.* The work to date has focused on the capture of contextual information in the form of an observation. However, taking data out into the field also serves a useful function [13]. The fieldwork environment could be augmented with a variety of information such as a flood warning message attached to a location and weather context, data from a past vegetation survey that needs to be updated in a particular area, etc.

Using these context-aware capabilities, the prototype can be extended beyond a simple data collection tool into a fully-fledged field-guide that can help the ecologist structure their work and access information that is relevant to their present situation from an 'encyclopedia' of knowledge about the area. Such developments will also benefit from a more advanced hardware system that can support a larger array of devices (e.g. digital cameras, magnetic compasses, range finding equipment, microphones, wireless communications etc.) in a more truly wearable configuration than we have attempted with this first prototype.

## 5. The context information service

The case study demonstrates that, even with modest hardware and contextual capabilities, great benefits can be derived. We believe this to be true of many application domains, not just fieldwork. Yet progress toward integrating context-aware capabilities into wearable computers and other computing environments is sluggish. This is despite the wide array of potential applications and the numerous application frameworks and sensor technologies available. For location contexts alone, there are a multitude of sensing technologies available such as GPS [5], active badges [14], piconets [15], etc. However, it may be that the sheer diversity of exploitable contexts and this plethora of sensing technologies are actually working *against* context-awareness. Creating software that can interact with a variety of hardware to capture a context, translate it to a desirable format, perform various manipulations and comparisons on it, and present it to the user in a meaningful way, is a difficult and time-consuming software development task. For example, to incorporate a location context into the prototype fieldwork tool, the software had to be able to communicate with the GPS device, extract the location information, convert it from latitude and longitude to a desirable format, and to project location data onto a map display. Any application that wishes to incorporate a location-awareness faces these same challenges, and more so for each new sensor device and context they choose to support. We believe that it is this prohibitively large development overhead that has stifled more widespread adoption and experimentation of contextual capabilities.

To remedy this situation we have begun work on a Contextual Information Service (CIS). This service maintains a dynamic model of contextual information, providing a common access medium to the current context for any program, service, or user, that is associated with the wearable computer. The aim of the CIS is to offer a common contextual model that can provide any client with any contextual information that they may require, with minimal effort on the client's part.

### 5.1 Modeling context

Context is a subjective concept that is defined by the entity that perceives it. For example, one entity may conceive of its context as location whereas another may view it from a temporal perspective. It could also be a more ethereal construct, e.g. the emotional state of a

person. Therefore, context could be generally described as the subset of physical and conceptual states of interest to a particular entity.

It would be possible to implement a simple contextual model in the form of a set of states that are automatically updated, e.g. a location state updated from a GPS, a time state updated from the internal clock, etc. However, contextual states do not exist in a vacuum, they are inherently associated with specific objects. For example, a location state by itself is purposeless unless attributed to a particular object. In the simple state model, assumptions could be made about what they relate to, e.g. location could be reasonably assumed to be the position of the computer. However, this precludes the possibility of modeling the state of more than one entity, e.g. location of the user and the location of their colleagues. For this reason the CIS model is object oriented, with states embedded in entities called *artifacts*.

The CIS model consists of a *world* that is comprised of artifacts. These artifacts in turn consist of a name, type, and set of contextual states. For example, the world may contain a person artifact named 'Jason' with a location state. A CIS client can access any state of any artifact in the world (e.g. Jason.Location), or can add their own artifact. Adding a new artifact is simply a matter of selecting a *template* from the *artifact catalog*, and giving it a name, e.g. selecting a person template and naming it 'Jason'. If a suitable template does not already exist in the catalog then a new entry, consisting of a type-name and set of states, can be added.

The CIS also contains a *state catalog* that lists the various generic states. Each state has one or more formats and types, methods to translate between them, and a set of operations. These elements are only visible when adding a state to the catalog or augmenting an existing one; normally the client will simply be interested in the generic level, e.g. working with a location rather than latitude and longitude specifically.

## 5.2 Monitoring context

The modeling infrastructure described thus far is connected to the real world through the *sensor array*. This array is populated with a customizable set of *sensors* and *synthesizers*. Sensors are programs that extract contextual data from a connected or embedded hardware device (e.g. extracting location from a GPS receiver) and synthesizers are programs that synthesize contextual data from other

artifact states (e.g. synthesizing a tide level context from a location and time state).

Contextual data from a sensor or synthesizer is routed to the appropriate state of a particular artifact by a *monitor*. The monitor is associated with a single artifact state, operating under a quality of service directive from that state, e.g. location must be accurate to 30 meters. The monitor utilizes the sensor that best matches the quality of service required, and will switch sensors if necessary to maintain that quality of service. The monitor also records a history of sensor behavior and uses this to optimize performance (e.g. minimizing the frequency with which it checks the GPS sensor whilst still maintaining the quality of service threshold) or to provide simulated values during temporary sensor failures (e.g. if the user walks through some woods that the GPS signals cannot penetrate).

The sensors and synthesizers simply provide a current contextual value. Monitors decide when to obtain/generate these values and perform simulations. This is because state behavior is artifact-dependent. For example, a person artifact needs to have their location updated a lot less frequently than a car artifact because of the slower potential rate of change.

## 5.3 Relating context

The final element of the CIS is the *artifact relationship*. These relationships consist of an *anchor* that roots the relationship, and a set of members that may have *explicit* links (i.e. ones that are explicitly defined by the user) and *inferred* links (i.e. ones that are inferred by the CIS). Relationships can offer additional contextual information or change how we interpret existing information, e.g. if a person was *in* a car then we would expect their location state to change more rapidly.

Relationships are created by selecting a template in the *relationship catalog* (similar in principle to the artifact catalog), naming it, and specifying an anchor, optional parameters, and the candidate artifact types. For example, a *CloseTo* relationship called 'Nearby Printers' with 'Jason.Location' specified as an anchor, 'ten meters' as the range parameter, and 'printers' as the only candidate artifact type. The resulting group of printers related to my location are automatically updated as I walk around, so that printers I leave behind are removed from the relationship and printers that I approach are included in the relationship. Relationships work on a generic level so that they can be applied to many different states and

artifacts, e.g. the 'CloseTo' relationship could equally be applied to temperature.

## 5.4 General characteristics

Issues of extensibility and reusability are paramount, as there are an infinite number of artifacts and states that a client may wish to observe. The CIS addresses these concerns by providing catalogs of components, any of which can be augmented as required. To model a particular context the client simply selects the appropriate components from a catalog, or, alternatively, can reuse artifacts that are already in the world. The latter case is a kind of run-time reusability where different clients have shared interest in particular parts of the model. Producing one optimized model rather than several application dependent ones also increases efficiency by eliminating any redundant or duplicated processing or programming.

## 6. Conclusion

Four generic contextual capabilities that can be utilized by wearable systems have been identified: sensing, adaptation, resource discovery, and augmentation. A selection of these techniques was incorporated into a wearable computer prototype designed to assist an ecologist's giraffe observations in Kenya. The prototype proved successful over a two-month trial, allowing the ecologist to complete more work in shorter periods, and the context-aware features were attributed as a critical part of the system's success. It is likely that many other applications can also benefit in similar ways. However, the complexity of obtaining and working with contextual data currently impedes the utilization of context-awareness in many applications. A brief overview has been given of our current work in designing and implementing a Contextual Information Service to remedy this situation. The CIS encourages developers to add context-aware features to their software by providing them with an extensible shared model of context that is transparent of any underlying complexities.

## 7. Acknowledgements

Many thanks to all the people that helped make the Kenyan trial possible. Special thanks go to Alan Birkett from Manchester Metropolitan University, who organized the expedition, and Kathy Pinkney who intrepidly used the equipment for her study of giraffe.

## 8. References

- [1] B. Schilit, N. Adams, R. Want, Context-Aware Computing Applications, *IEEE Workshop on Mobile Computing Systems and Applications*, 85-90, 1994.
- [2] S. Feiner, B. MacIntyre, T. Höllerer, and A. Webster, A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment, *1st International Symposium on Wearable Computers (ISWC '97)*, October 1997.
- [3] S. Long, R. Kooper, G. D. Abowd, and C. G. Atkeson, Rapid Prototyping of Mobile Context-Aware Applications: The Cyberguide Case Study, *Proceedings of the 2nd ACM International Conference on Mobile Computing and Networking (MobiCom'96)*, November 1996.
- [4] R. Hull, P. Neaves, J. Bedford-Roberts, Towards Situated Computing, *1st International Symposium on Wearable Computers (ISWC '97)*, October 1997.
- [5] Global Positioning System Overview, Available at <<http://www.utexas.edu/depts/grg/gcraft/notes/gps/gps.html>>
- [6] S. M. Thompson, D. E. Sheat, Exploiting Telecommunications to Deliver Real Time Transport Information, *IEE 9th International Conference on Road Traffic Information & Control*, forthcoming.
- [7] M. Weiser, Some Computer Science Issues in Ubiquitous Computing, *Communications of the ACM*, 1993, 36(7), 74-85.
- [8] G.W. Fitzmaurice, Situated Information Spaces and Spatially Aware Palmtop Computers, *Communications of the ACM*, 1993, 36(7), 38-49.
- [9] 'Mobile Computing in Fieldwork Environments' project homepage, Available at <<http://www.cs.ukc.ac.uk/research/infosys/mobicomp/Fieldwork/index.html>>.
- [10] J. Pascoe, D. R. Morse, N. S. Ryan, Developing Personal Technology for the Field, *Personal Technologies*, submitted for publication.
- [11] P. J. Brown, J. D. Bovey, X. Chen, Context-aware Applications: from the Laboratory to the Marketplace, *IEEE Personal Communications*, 4(5), 58-64, 1997.
- [12] J. Pascoe, N. S. Ryan, D. R. Morse, Human-Computer-Giraffe Interaction – HCI in the Field, *Workshop on Human Computer Interaction with Mobile Devices*, University of Glasgow, United Kingdom, May 1998.
- [13] N. S. Ryan, J. Pascoe, D. R. Morse, Enhanced Reality Fieldwork: the Context-aware Archaeological Assistant, in V. Gaffney, M. van Leusen and S. Exxon (eds.) *Computer Applications in Archaeology 1997*, forthcoming.
- [14] R. Want, A. Hopper, V. Falcao et al, The Active Badge Location System, *ACM Transactions on Information Systems*, 10(1), 91-102, 1992.
- [15] F. Bennett, D. Clark, Piconet: Embedded Mobile Networking, *IEEE Personal Communications*, 4(5), 1997.