

Exploiting Context in HCI Design for Mobile Systems

Tom Rodden, Keith Chervest, Nigel Davies,
Department of Computing, Lancaster University, Lancaster, LA1 4YR.

Alan Dix,
School of Computing, Staffordshire University, Stafford, ST18 ODG.

Background

The last five years has seen a shift in the nature of mobile computers. The development of increasingly powerful laptop computer systems has been mirrored by the production of a range of small computational devices. The increased prominence of these devices outlined a number of distinct research challenges. These challenges have tended to focus on extending the utility of these devices using new forms of interaction; techniques to overcome display limitations or improvements in the general ergonomics of these devices. The merging of these devices with existing telecommunication services and the production of devices that offer connections to other systems presents yet another set of research challenges in terms of the development of cooperative multi-user applications.

The authors are engaged in a number of projects investigating various aspects of mobile systems development. In particular, an MNA funded project "Interfaces And Infrastructure For Mobile Multimedia Applications" is looking at the way in which the special user interface requirements of cooperative mobile systems can be used directly to drive the development of an effective system architecture, user interface toolkit and underlying communications infrastructure.

In various ways mobile systems break assumptions that are implicit in the design of fixed-location computer applications leading to new design challenges and feeding back to a better understanding of the richness of human-computer interaction.

One central aspect of our work is the temporal issues that arise due to network delays and intermittent network availability. We have already addressed this in some detail based on previous theoretical work on pace of interaction and practical experience in building collaborative mobile applications [Dix, 1992; Davies 1994; Dix, 1995]. In addition, there has been considerable wider interest in temporal issues, both in the context of mobile systems and also more generally [Johnson, 1996; Johnson, 1997; BCSHCI, 1997; Howard and Fabre, 1998].

However, this paper considers a second critical issue in the design and development of cooperative mobile systems, the context sensitive nature of mobile devices. the importance of this is clear in the recent research in ubiquitous computing, wearable computers and augmented reality [Weiser, 1991, 1994; Aliaga, 1997]. Furthermore, more prosaic developments such as mobile phones, GPS and embedded in-car automation all point to a more mobile and embedded future for computation. This development of applications which exploit the potential offered by this technology brings together issues from distributed systems, HCI and CSCW. However, designers of these systems currently have few principles to guide their work. In this paper we explore the development of a framework that articulates the design space for this class of system and in doing so points to future principles for the development of these systems.

Fixed-location computers are clearly used for a variety of tasks and are set within a rich social and organisational context. However, this is at best realised within individual applications and the nature of the device as a whole is fixed and acontextual. In contrast, the very nature of mobile devices sets them within a multi-faceted contextual matrix, bound into the physical nature of the application domain and closely meshed with existing work settings. In this paper we seek to articulate the nature of this matrix and how it may be used as a resource for designers and developers.

Making use of the context of a device is important for two reasons. Firstly, it may allow us to produce new applications based on the special nature of the context, for example interactive guide maps. Secondly, and equally important, it can help us tailor standard applications for mobile devices, for example when a sales rep visits a company, the spreadsheet can have a default files menu which includes the recent ordering history for the company. Such tailoring is not just an added extra, limited screen displays mean that highly adaptive, contextual interfaces become necessary for acceptable interaction.

Moving from the device to the context of use

A considerable amount of research surrounding the development of mobile devices has obviously focused on the portable nature of these devices and the technical problems in realising these. Mobile computing devices represent real technical challenges and have always stretched the state of the art in terms of displays and interaction devices.

This focus on the development of appropriate forms of device is perhaps best exemplified by the development of so called "wearable computers". These have seen the construction of new forms of interaction devices that support a limited number of dedicated tasks. These have included support for mechanics, portable teaching aids and note taking machines [Fickas 1997].

The development of dedicated function devices is complemented by the emergence of a range of general purpose devices are normally characterised as Personal Digital Assistants. The majority of these devices focus on supporting some form of personal organisation by combining diary and note taking facilities. These devices are characterised by their personal and individual nature and any communication provided has focused on providing support for access to on-line information such as the email and the World Wide Web.

The emergence of mobile telecommunication standards such as GSM and the increased availability of these services has also led more recently to the development of a range of devices that provide mobile access to on-line services (e.g., the Nokia communicator). This merging of computer and communication facilities allows the development of systems that provide on-line immediate access to information. These portable networked devices have also been combined with the use of GPS technologies to develop a range of portable devices that are aware of their position [Long 1996].

The ability of the current generation of portable devices to have an awareness of their setting and an increased ability to access network resources means that we need to broaden our consideration of these devices to see their use in tandem with other portable devices. This view of portable devices means that we need to balance the current consideration of the interaction properties of individual devices with a broader consideration of the context of use. This move toward a consideration of the context of use builds upon previous trends in the development of portable devices, includes the use of TABS in developing mediaspaces at PARC and the associated emergence of the notion of Ubiquitous computing [Weiser, 1991, 1993]. More recent work at MIT has also focused on the development of small-scale devices that exploit context to provide an ambient awareness of interaction [Ishii 1997].

Considering the context of Mobile Systems

Our particular focus is a consideration of applications that we term *advanced mobile applications*. Although research prototypes exist that demonstrate the technical possibilities many of these have yet to emerge as fully-fledged applications. These applications are distributed in nature and characterised by peer-to-peer and group communications, use of multimedia data and support for collaborating users. Examples of such applications include mobile multimedia conferencing and collaborative applications to support the emergency services.

In considering the design and development of interfaces for mobile devices we wish to particularly focus on the situation where mobile devices behave differently and offer different interaction possibilities depending on the particular context in which the system is been used. For example, in the development of mobile multimedia guides such as the systems at Georgia Tech [Long 1996] and the Lancaster Guide [Davies 1998] the information presented to the user and the interaction possibilities is strongly linked to the location where the device is been used. Interaction is no longer solely a property of the device but rather is strongly dependant on the context in which the device is been used.

In this paper we wish to examine the nature of the context in which mobile devices are used and the implications for future HCI design. The aim of this focus on the context is to allow the highly situated nature of the devices to be reflected in the design of interactive systems that exploit these systems. This focus on the situated nature of these devices reflects the growing acceptance of these devices and the need to allow them to closely mesh with the existing practices. This needs to focus on the context of use mirrors previous work in the development of interactive systems within CSCW [Hughes 1994]

In considering context as a starting point for the design of interaction within we need to unpack what we actually mean by the term context and how we may exploit to determine different interaction possibilities within mobile systems. The following sections consider some of the ways in which context has played a key design role in the development of distributed mobile applications and the consequences suggested for the development of future applications.

Infrastructure Context

The interaction offered by advanced mobile applications is not solely dependent on the particular features of the mobile devices used. Rather it is a product of the device and the supporting infrastructure used to realise the application. The impact of the properties of the supporting distribution infrastructure to different styles of interaction has been discussed in CSCW and HCI [Greenberg 1994]. In mobile systems the nature of the infrastructure is even more likely to change as the application is used and the sort of service available may alter dramatically. This

variability in the infrastructure may dramatically effect interaction and it is essential that interaction styles and interfaces provide access to information reflecting the state of the infrastructure

This issue is particularly acute in the case of safety critical applications where applications must be rigorously engineered to ensure a high level of dependability. The dependency of these systems comes not only from the reliability of the communication infrastructure and devices but also from the users awareness of the nature of the application. Provision of this awareness require us to reconsider the traditional views of distribution transparency and abstraction and allow the user access to the properties of the infrastructure and infer different interaction results from this contextual information.

In essence, the user interfaces to mobile applications must be designed to cope with the level of uncertainty that is inevitably introduced into any system that uses wireless communications. For example, consider our experiences in the development of an advanced mobile application used to support collaborative access to safety critical information by a group of field engineers [Davies 1994]. If one of these engineers becomes disconnected from the group as a result of communications failure then it is vital that the remaining user's interfaces reflect this fact. This requires interaction between the application's user interface and the underlying communications infrastructure via which failures will be reported. In addition, if the information being manipulated is replicated by the underlying distributed systems platform the validity of each replica will clearly be important to the engineers. In this case the user interface will need to reflect information being obtained from the platform.

The design of these applications needs to not only reflect the semantics of the application and the features supported but it must also consider as a key design element the variability of the supporting infrastructure and how this variability is reflected to the user. Similarly, the particular features of the infrastructure may need to be put in place and designed in line with the interaction needs of the mobile application.

Application Context

In addition to the infrastructure issues discussed above, distributed mobile applications need to consider the detailed semantics of the application. In the case of mobile applications the normal design considerations are amplified by the need to consider the limited interaction facilities of mobile devices. In addition, a number of additional contextual issues need to be considered in the design of these applications.

Mobile devices are intended to be readily available and of use to the community of users being supported. As a consequence we need to consider the highly situated nature of this interaction. Developing a clear understanding of what people do in practice and the relationship with technology is essential to informing the development of these applications. The relationship between users and mobile technology is still unclear and few studies have taken place that considers the development of mobile cooperative application [Davies 1994].

For example we may choose to exploit the personal nature of these devices to associate mobile devices with users. This allows us to tailor applications to allow them to be sensitive to the identity of the user of the device. This information may be exploited along with additional contextual information (e.g. location) to present appropriate information. One example of this would be a particular doctor visiting patients within a hospital. At a particular bed, who the doctor is and their relationship to the patient in the bed may determine the information presented. Contrast this situation with the development of a museum guide where the devices need to be considered as general purpose and no information is available about the relationship between users and the artefact being described.

The design of advanced multimedia applications needs to explicitly identify the nature of the work being supported and the practicalities of this work. In doing so developers need to consider the relationship between the mobile devices and their users and how this can be used to determine the nature of the interfaces presented. This is particularly important if devices are to be used to identify users and potentially make information about their location and what they are doing available to others. In this case a consideration of the issues of privacy and the need for some symmetry of control is essential.

System Context

In addition, to exploiting information about who will be using devices, interaction with mobile applications also needs to consider the system as a whole. The nature of these devices is that more advanced applications need to be distributed in nature. Thus rather than having functionality reside solely within a single machine (or device) it is spread across the system as a whole. This means we need to consider the interaction properties of the system in terms of the distributed nature of the application. This is particular true when we consider issues of pace and interaction [Dix, 1992]. Consider for example, the development of appropriate caching strategies for field engineers who will only ever be examining or servicing units within a sub region of a particular area.

The need for rapid feedback is an accepted premise of HCI design and many applications provide direct manipulation interfaces based on the ability to provide rapid feedback. The development of distributed applications

has seen a reconsideration of the nature of feedback and the importance of considering the technical infrastructure as impacting this [Dix, 1995]. The variable nature of the Internet and the effects on World Wide Web interaction is perhaps the most readily identifiable manifestation of this effect [BCSHCI, 1997]. A natural design tension exists between replicated application architectures that maximise feedback and centralised applications that prioritise feedthrough across the application users [Ramduny and Dix, 1997]. The need to consider the overall functionality of the application and to design structures that provide appropriate access to different levels of functionality is amplified in the case of mobile applications where the infrastructure may vary considerably as the application is in use.

Location Context

One of the unique aspects of mobile devices is that they can have an awareness of the location within which they are being used. This location information may be exploited in determining the form of interaction supported. This may either be direct in that it explicitly exploits the nature of the setting within the application. For example, the development of guides that tell you about your current location. It may also be less direct in the development of systems that inform you of incidents depending on your particular location.

The degree to which the mobile application is coupled with the location of devices and how this location is made available to users is a key design decision in supporting different interaction styles. The device used and the form of interaction it supports is not the sole determinant in the form of interaction. Rather it is a product of the location of the devices and the location of other devices. This means that we need to consider the issues involved in the correspondence between these devices and their location. For example, if a guide describes a particular location and is dependent on references to that location to support the interaction we must ensure that this contextual reference is maintained. It is essential that our approaches to design explicitly involve the issues of location and the link with these contextual cues.

Physical Context

Finally, mobile computer systems are likely to be aware of, or embedded into their physical surroundings. Often this is because they are embedded in an application specific device, for example in a mobile phone or car. In these situations the computer system is mobile by virtue of being part of a larger mobile artefact. This context can and does affect the application interface, for example, the telephone directory within a mobile phone can be very different from one in an independent PDA. Another example is a car radio (now often computer controlled) which has different design considerations to a static radio including the need to automatically retune as the car travels between local radio areas and transmitter zones. Because the computer systems are embedded into application specific devices they may also be aware of their *environmental context*, for example, the speed of the car. Some of this sensory information may be used simply to deliver information directly to the user, but others may be used to modify interface behaviour. For example, in a tourist guide, increasing text size in poor lighting conditions or, in a car system, limiting unimportant feedback during periods of rapid manoeuvring.

Each of these different contexts represent different portions of the design space within which mobile systems must be placed and the features of infrastructure, application, system and location all provide potential trade-offs that developers must address in realising mobile interactive systems. Currently, designers undertake this trade off with little support or guidance as the system as little is known of the extent of the design space into which mobile applications are placed. In the following section we wish to consider a taxonomy of mobile computation that charts this design space to allow developers to consider the properties of the mobile system under construction and how this may be related to other applications and systems.

Towards a taxonomy of mobile computation

Having considered some of the different ways in which context may affect or be used in mobile devices; we now want to build a classification of mobile and context-aware devices to better understand the design space. Clearly, as we are considering mobile systems, ideas of space and location are of paramount importance in our consideration of the context of these systems. We will therefore first examine different kinds of real and virtual location and different levels of mobility including issues of control. However any notion of location puts the device within an environment which has both attributes itself and may contain other devices and users with which the device may interact.

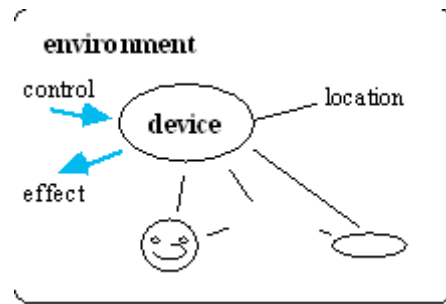


Figure 1. A device in its environment

Of real and virtual worlds

A lawnmower is a physical device; it inhabits the real world and can be used to affect the real world. Computers open up a different kind of existence in an electronic or virtual world. This is not just the realm of virtual reality, as we surf the web, use ftp to access remote files or even simply explore our own file system we are in a sense inhabiting virtual space. Even the vocabulary we use reflects this: we 'visit', 'explore', 'go to', 'navigate' ... our web browsers even have a button to go 'back'. There has been a growing acceptance of the consideration of a virtual space and the development of electronic worlds and landscapes.

The emergence of virtual space

The turn to virtual worlds and spatial approaches generally has emerged from work in HCI and CSCW on the use of spatial metaphors and techniques to represent information and action in electronic systems. This work has its roots in the use of a rooms metaphor to allow the presentation of information [Henderson, 1985]. From these early spatial approaches we have seen concepts of spatial arrangement exploited in the development of desktop conferencing systems such as Cruiser [Root, 1988] and more generally in the work of Mediaspaces [Gaver, 1992].

The recent development of co-operative systems in CSCW has also seen a growing application of concepts drawn from spatial arrangements. These include the development of *groupkit* to form *teamrooms* [Roseman, 1996], the emergence of the *worlds* system [Fitzpatrick, 1996] and the use of a notion of places to support infrastructure [Patterson, 1996]. This exploitation of virtual spaces is most notable in the development of shared social worlds existing solely within the machine [Benford, 1995]. However, the use of space and virtual spaces has not been isolated to an existence solely within the computer and a number of researchers have considered how space and location can be considered in both virtually and physically within the development of applications. This is most evident in the augmenting of existing physical spaces to form digital spaces populated by electronically sensitive physical artefacts (or tangible bits) [Ishii, 1997] that are sensitive to their position within both physical and virtual space.

Combining the real and the virtual

The work in tangible bit undertaken by Ishii (1997) represents the start of a trend to interweave real and virtual spaces that exploit a capability differently offered by mobile computer applications and we would suggest that this interplay between the real and the virtual is at the core of the design of co-operative mobile applications as devices and users have a location and presence that is both virtual and physical each of which is available to the computer application.

This interplay between the real and the virtual provides a starting point for the development of our taxonomy. A direct result of the need to recognise this coupling is that many of the categories we will consider for taxonomising and understanding mobile and context-aware computation have counterparts in both the real physical world and virtual electronic world. There are important differences and the virtual world does not always behave in ways we have come to expect from the physical world and these differences are often exploited by designers and developers.

In particular, even the object of interest for mobile computation may have a physical or virtual existence depending on the nature of the application. At one extreme we have simple hand held GPS systems that simply tell you where you are in physical space – perhaps these do not even rank as mobile computation. At the other extreme there are agents which simply have an existence within the virtual world, for example web crawlers or the components within CyberDesk [Wood, 1997]. Between these we have more complex physical devices such as the PDA which have both a real world existence and also serve as windows into virtual space (especially when combined with mobile communications).

In the development of the taxonomy presented here we will focus will on physical mobile computational devices. However, we will also draw on examples of virtual agents where they are instructive to highlight the co-existence of these two forms of space and the issues of mobility that may exist in both.

Location

Mobility makes us think of automatically about location and the way in which this sense of location can be both understood in the system and changes in location can effect the system. Any simple mobile device will have a physical location both in space and time. Understanding the nature of this location and how the developers of interactive mobile applications may exploit it is important and in this section we wish to consider what we might actually mean by the term location. This exploration is more than a mere issue of terminology as developing a understanding of what we actually mean by location represents a consideration of one of the core design concepts in the production of mobile systems.

Looking at the spatial dimension there are some devices (for example GPS based map systems) where the exact Cartesian position in 2D or 3D space is important in defining a sense of absolute physical location. For others a more topological idea of space is sufficient in understanding position and in these case location is consider not in an absolute sense but in relation to other objects or sensors. For example the Lancaster GUIDE system is based on radio cells roughly corresponding to rooms and sections of Lancaster Castle and the CyberGuide [Long,1996] system at Georgia Tech. shows visitors around the GVV laboratory by altering its behaviour depending on what item of equipment is closest.

This distinction between a sense of the absolute and relative in location can also be applied to time. We can consider a simple, linear, Cartesian time typified by a scheduler or alarm clock. However, we can also have applications where a more relative measure of time is used, for example, during a soccer match we may consider the action in the first half, second half and extra time but not care exactly whether the match was played at 3pm or 6pm. Similarly, in the record of a chess game, all that matters is the order of the moves, not how long they took. In fact, many calendar systems employ a hierarchical and relative model of time: hours within days, days within weeks. At first this might seem like a simple division of linear time, but such systems often disallow appointments spanning midnight, or multi-day meetings that cross into two weeks.

We can thus think of both space and time as falling into 'Cartesian' and topological categories and can consider location in both space and time in these terms. We may also consider location in both a physical and virtual sense. If we consider ideas of virtual location, for example position within a hypertext, we see that we may similarly have ideas of time and space within the electronic domain. As an example of virtual time consider looking up next week's appointments in a scheduler, the real time and virtual time need not correspond. For those with busy schedule these seldom correspond and the art of mapping from the real to the virtual is often an delicate balancing act worked out in practice.

This consideration of location provides us with the following categorisation of location:

		real	Virtual
space	Cartesian topological	GPS room	VR hypertext
time	linear topological	stop watch calendar	history time line

Figure 2. Location in different kinds of space

Note that these are not mutually exclusive categories: an item in a room also has a precise longitude and latitude, a device will exist at a precise moment in linear time, but may being used to view past events at that stage. Indeed possibly one of the most interesting things is where these different ideas of location are linked to allow visualisation and control. For example, moving a display up and down in physical space could be used to change the virtual time in an archaeological visualisation systems, and in an aircraft cockpit, setting the destination city (topological destination) instructs the autopilot to take an appropriate course in Cartesian space/time. This interplay between the real and the virtual is central to the development of augmented reality spaces where the movement of devices within a space may manifest in effects that are both real and virtual. These spaces only work because the location of the device can be controlled in virtual and physical space and its effects proved alterations to either the physical or virtual space.

Mobility

Out core concern in the development of our design taxonomy is the issue of mobility and its implication for how we understand human computer interaction. In the previous section we considered how the issue of location can be unpacked to provide an understanding in both a physical and virtual sense and how the nature of the space effects our

consideration of location. In this section we wish to focus on how might understand mobility and what potential design issues may emerge from a more detailed consideration of mobility.

Devices may be mobile for a number of reasons. They may be mobile because they are carried around by users (as with a PDA or a wearable computer), because they move themselves (robots!) or because they are embedded within some other moving object (a car computer). Furthermore a number of different devices may be spread within our environment so that they become pervasive, as in the case of an active room such as the ambient room suggested by Ishii(1997). The issue of pervasive is itself a rather thorny one in that we it is not clear what constitutes pervasive in terms of devices and how this relates to previous discussions surrounding ubiquitous devices. The issue of ubiquitous computing has focused on the backgrounding of the device and the computer essentially "disappearing" into the environment. For us the issue of pervasive device has less to do with the devices fading into the environment and more to do with an expectation that particular devices are normally available. For us pervasive computing is intimately bound up with the inter-relationship between different devices and the expectation that these devices can work in unison to provide some form of shared functionality. An active room is active because it contains a number of devices which when they work in unison provide some form function. Essentially, we are seeing a number of computing devices working in co-operation to provide some functionality and some of these devices may be mobile. However, often these devices are not. Consider for example the layout of base stations that provide the information displayed on mobile devices to allow a space to offer some form of pervasive computing facility.

We can disentangle the different levels of mobility into three dimensions which are used in Figure 3 to classify example mobile systems.

First we can consider the level of mobility within the environment:

- fixed – that is the device is not mobile at all! (e.g a base station fixed in a particular place)
- mobile – may be moved by others (e.g. carried around, e.g PDA or wearable computer)
- autonomous – may move under its own control (e.g. a robot)

Second, we can consider the extent to which the device is related to other devices or its environment:

- free – the computational device is independent of other devices and its functionality is essentially self contained.
- embedded – the device is part of a larger device
- pervasive – the functionality provided by the device is essentially spread throughout the environment and results for the a devices relation to other elements in the environment.

These separations do not consider the nature of the device and the sort of functions it may afford. The physical design of the device itself is an issue that needs to be considered carefully and needs to be considered in terms of existing traditions of aesthetic and practical design. The consideration of these features are beyond the scope of the framework and taxonomy we wish to present here which focuses on the development of the device.

As a final part of our taxonomy we can reflect the co-operative nature of advanced mobile applications by considering the extent to which the device is bound to a particular individual or group. We have three classes for this too:

- personal – the device is primarily focused on supporting one person
- group – the device supports members of a group such as a family
- public – the device is available to a wide group

We would not suggest that these categories are absolute but rather provide them as sample equivalent cases of utility to designers. All the categories have grey cases, but perhaps this last dimension most of all. In particular we should really consider both the static and dynamic nature of how these categories are applied. For example, we could classify a computer laboratory as 'public', but of course, after logging in, each computer becomes personal. We will return to these dynamic aspects when we look at how devices can become aware of their users.

In fact, the 'group' category really covers two types of device. Some, like a liveboard actually support a group working together. Others, like an active refrigerator (which allows messages to be left, email browsing etc.) may primarily support one person at a time but is available to all members of a family. In-car computers systems exhibit both sorts of 'groupness', they may perform functions for the benefit of the passengers of the car as well as the driver and also the exact mix of people from within the family (or others) within the car may vary from trip to trip.

Some of the examples in Figure 3 are clear, but some may need a little explanation. The 'Star Trek' reference is to the computer in Star Trek that responds to voice commands anywhere in the ship, but does not actually control the

ship's movements. This is perhaps a wise move given the example of HAL in the 2001! (Note HAL is put in the group category as it has a small crew, but this is exactly one of the grey distinctions.) Our reference to 'shopping cart' refers to the development of supermarket trolleys that allow you to scan items as they are added and keeps track of your purchases to enable a fast checkout. Often these require the insertion of a shopper identification, in which case these become dynamically personalised.

		Personal	Group	public
	Fixed	office PC	Liveboard	Computer lab.
Free	Mobile	PDA	tour guides	
	Autonomous			Factory robot
	Fixed		Active fridge	ATM
Embedded	Mobile	Wearable devices	Car computer	Shopping cart
	Autonomous		Auto pilot	Mono rail
	Fixed		active room	
Pervasive	Mobile			Star Trek
	Autonomous	web agent	HAL	web crawler

Figure 3. A Taxonomy of different levels of mobility

Notice there are various blank cells in this taxonomy reflecting our use of the taxonomy as a means of charting the design space for interactive mobile devices. Some of these blanks represent difficult cases where there may not be any sensible device. For example, a fixed–pervasive–personal device would have to be something like an active hermits cell. In fact, the whole pervasive–personal category is problematic and the items 'web agent' and 'web crawler' in the final row may be better regarded as virtual devices of the free–autonomous class.

Other gaps represent potential research opportunities. For example, what would constitute a free–mobile–group device? This would be a portable computational device that supports either different individuals from a group, or a group working together – possibly an electronic map that can be passed around and marked.

Most of the examples are of physical devices. Virtual devices may also be classified in a similar way, for example, Word macros are embedded–mobile (or even autonomous in the case of macro viruses!) as are Java applets. The only virtual devices in Figure 3 are the items 'web agent' and 'web crawler' in the final row. However, these are perhaps better regarded as virtual devices of the free–autonomous class. This ambiguity is because any virtual device or agent must be stored and executed upon a physical computational device and the attributes of the physical device and virtual device may easily differ. For example, a PDA may contain a diary application. This is mobile by virtue of being stored within the PDA (a virtual device embedded within a physical device). However, if the PDA is used as a web browser it may execute a Java applet that is a form of virtual agent embedded within a web page (a virtual embedding in a mobile artefact). That is we have an embedded–mobile–public virtual agent temporarily executing on a free–mobile–personal device! This dual presence in multiple contexts is both the difficulty and the power of virtual environments and one that requires some significant research to resolve.

Populating an environment

Devices may need to be aware of aspects of their environment in addition to their location within it. These may vary because the device is moving from location to location (the headlamps on a car turning on automatically as the car goes into a tunnel) or because the environment is changing (a temperature monitor). In a sense, devices need to be aware that they populate an environment and need to reflect the coupling with the environment depicted in Figure 1.

This awareness may include both the physical environment (light, temperature, weather) and the electronic environment (network state, available memory, current operating system). A simple of the latter are Javascript web pages which run different code depending on the browser they are running on.

Environments are normally populated with a range of different devices. Within the physical and virtual environment of a device there may be other computational devices, people (including the user(s) of the device) and passive objects such as furniture. These may be used to modify the behaviour of the device. For example, in CyberDesk 'ActOn' buttons are generated depending on what other applications are available and the types of input they can accept.

Figure 4 gives examples of items in the environment that may be relevant for a mobile or context-aware device taking a car computer and an active web page as running examples.

	Physical	Virtual
People	Current driver of car	visitor at web page

Devices	other cars	running applets
Objects	roadside fence	other pages on the site

Figure 4. Examples of entities within the environment

This sense of awareness of the surrounding environment and conveying this awareness to others is an issue of some sensitivity in design. For example, in the case of active badges the issue of awareness of users and how this may be applied became embroiled within a discussion of privacy [Harper, 1992]. This may become even more problematic in the case of multiple devices that display an awareness of others. For example, consider the suggested "fun" interest badge device offered by Philips in the development of its visions of the future [Philips, 1996] design study. These badges are programmed with a set of interest profiles for people and are intended to light up when you meet someone else with a compatible profile. The social acceptability of this form of device may well become a significant issue in determining their acceptability and the general acceptance of devices of this form.

Measurement and awareness

In order to modify their behaviour devices must be able to detect or measure the various attributes we have mentioned: their location, environment, other devices, people and things.

These are mostly status phenomena and elsewhere [Dix and Abowd 1996, Ramduny, Dix and Rodden 1998] we have discussed the various ways in which an active agent can become aware of a status change. In short, these reduce to finding out directly or via another agent (human or electronic). For example, a car with a built in GPS sensor can detect its position directly and thus give directions to the driver, but a simple PDA may need to be told of the current location by its user in order to adjust timezones. Other computational agents may also be important sources of information about themselves (as in the case of CyberDesk) and about other parts of the environment (for example recommender systems).

Items in the environment (people, devices, objects) are particularly difficult: not only may they change their attributes (position etc.), but also the configuration of items may change over time (e.g. people may enter or leave an active room). This leads to three levels of awareness. We'll look at these with the example of a car computer:

- presence – someone has sat down in the driver's seat, but all the car can tell is that the door has been opened then closed
- identity – the driver enters her personal pin number and the car can then adjust the see position for the driver
- attributes – the car detects from the steering behaviour that the driver is getting drowsy and sounds a short warning buzzer

Notice how in this example, presence was not detected at all, identity was informed by the driver, but the sleepiness of the driver was detected directly. In other cases different combinations of detection or informing may be found. Security systems often have ultrasonic sensors to tell that someone is near (presence). Similarly, the car could be equipped with a pressure sensor in the driver's seat. Active badges, video-based face recognition or microphones matching footstep patterns can be used to tell a room who is there and hence play the occupant's favourite music and adjust the room temperature.

These examples are all about detecting people, but the same things occur in other settings. In the virtual world an agent may need to detect the same things: presence – whether any other applications are running, identity – if so what they are (e.g. Netscape), and attributes – what web page is currently being viewed. Also physical devices may detect one another for example allowing several people with PDAs to move into 'meeting' mode. Infact, awareness models that do just this form of detection within the virtual world abound[Rodden, 1996].

Detection and measurement may vary in accuracy: perhaps a box was put onto the car seat pressure sensor, the driver lied about her identity, the ultrasonic sensor cannot tell whether there is one or more people. It will also typically have some delay, especially when indirect means are used which is especially problematic if the attribute being measured changes rapidly. Thus actual detection is a trade-off between accuracy, timeliness and cost. Depending on the outcomes certain adaptations may be ill advised – a car wrongly identifies its driver and adjusts the seat thinking the driver is short, the real driver is quite tall and ends up squashed behind the steering wheel). The fidelity of awareness is very closely tied to the demands of the application and represents a genuine trade-off between the cost of measurement, the nature of the measurement and the importance of accuracy in the awareness information.

From requirements to architecture

As we have seen the taxonomy we suggest offers up many exciting design possibilities for specific applications suggested by the contextual nature of mobile devices. Although we are investigating some of these in a number of

projects at Lancaster University the primary aim of our current 'infrastructure' project is to examine the generic requirements to emerge from taxonomies of this form. These requirements can then be exploited to develop the underlying toolkits, architecture and infrastructure needed for temporally well designed, context-aware, collaborative mobile systems. One of the issues suggested strongly by our framework is that the issues of human computer interaction involved in mobile systems extend well beyond the interface provided by the device and have significant impacts on the infrastructure.

Research has demonstrated the shortcomings of existing infrastructure components for supporting adaptive mobile applications [Davies, 1994], [Joesph, 1995]. In more detail, existing components have two critical shortcomings. Firstly, they are often highly network specific and fail to provide adequate performance over a range of network infrastructures (e.g. TCP has been shown to perform poorly over wireless networks [Caceres, 1994]). Secondly, existing components often lack suitable APIs for passing status information to higher levels. As a consequence of these shortcomings new systems are increasingly being developed using bespoke communications protocols and user interfaces. For example, the GUIDE system described in [Davies 1998].

As these devices become more widespread the need increases for generic application architectures for at least subclasses. There is clear commercial pressure for this, in particular, Windows-CE is being promoted for use in embedded systems. However, if these are simply developed by modifying architectures and toolkits originally designed for fixed environments there is a danger that some of the rich interaction possibilities afforded by mobile devices may be lost.

There are some examples of generic frameworks on which we can build. In Georgia Tech., location aware guides are being constructed using the CyberDesk/Cameo architecture [Wood *et al.*, 1997]. Cameo is a software architecture based on the theoretical framework of status–event analysis. Status–event analysis lays equal weight to events, which occur at specific times, and status, phenomena which always have a value which can be sampled [Dix and Abowd, 1996]. The discrete nature of computation forces and emphasis in many specification and implementation notations towards the former, however, most contextual information is of the latter type – status phenomena. Computation using status phenomena requires callback-type programming, as is familiar in many user interface toolkits, to be used far more widely.

Another major architectural issue for context-aware applications is the way in which contextual issues cut across the whole system design. This is reminiscent of other aspects of user-interface where the structures apparent at the user interface often do not match those necessary for efficient implementation and sound software engineering [Dix and Harrison, 1989]. In UI design this has led to a conflict between architectures which decompose in terms of user interface layers, such as the Seeheim and ARCH-Slinkey models [Gram and Cockton, 1996] and more functionally decomposed object-oriented models. In fact the object and agent-based architectures themselves usually include a layered decomposition at the object level as in the MVC (Model–View–Controller) model [Lewis, 1995] and in the PAC (Presentation–Abstraction–Control) model [Coutaz, 1987]. Although the display and input hardware may be encapsulated in a single object or group of objects, its effects are felt in the architectural design of virtually every user-interface component. In a similar fashion the hardware that supplies contextual information may well be encapsulated within context-objects, but their effect will permeate the system. This requires a similar orthogonal matrix structure to that found in models such as PAC or MVC.

Conclusion

In this paper we have considered human computer interaction with mobile devices in terms of the development of advanced mobile applications. The maturing of technology to allow the emergence of multi-user distributed applications that exploit mobile applications means that we can no longer focus the issues of interaction on the nature of the device. Rather we must explicitly consider impact of the context in informing the design of different interaction techniques. The context needs to be considered in terms of the devices relationship with the technical infrastructure, the application domain, the socio-technical system in which it is situated, the location of its use and the physical nature of the device. The interaction style supported by this class of mobile application is as dependant on this context as the properties of the device itself. As a result, it is essential that work on the nature of these devices and the development of techniques that are aware of the limits of these devices is complemented by a broader consideration of the nature of interaction. However, these modified and novel forms of interaction cannot be realised without corresponding software architectures. So far we have identified two major structural principles which underlie this architectural design: the importance of representing status phenomena and the need for contextual information to cut across the software design space.

References

- Aliaga, D. G. (1997). Virtual objects in the real world. *Communications of the ACM*, **40**(3): 49-54.
- BCS HCI (1997). British HCI Group Workshop on *Time and the Web*. Staffordshire University, June 1997.

- Benford, S., Bowers, J., Fahlen, L., Mariani, J, Rodden. T, Supporting Cooperative Work in Virtual Environments. *The Computer Journal*, 1995. 38(1).
- Cáceres, R., and L. Iftode. "The Effects Of Mobility on Reliable Transport Protocols." *Proc. 14th International Conference on Distributed Computer Systems (ICDCS)*, Poznan, Poland, Pages 12-20. 22-24 June 1994.
- Coutaz, J. (1987). PAC, an object oriented model for dialogue design. *Human-Computer Interaction INTERACT'87*, Eds. H.-J. Bullinger and B. Shackel. Elsevier (North-Holland). pp. 431-436.
- Davies, N., G. Blair, K. Cheverst, and A. Friday. "Supporting Adaptive Services in a Heterogeneous Mobile Environment." *Proc. Workshop on Mobile Computing Systems and Applications (MCSA)*, Santa Cruz, CA, U.S., Editor: Luis-Felipe Cabrera and Mahadev Satyanarayanan, IEEE Computer Society Press, Pages 153-157. December 1994.
- Davies, N., K. Mitchell, K. Cheverst, and G.S. Blair. "Developing a Context Sensitive Tourist Guide", *Technical Report* Computing Department, Lancaster University. March 1998.
- Dix, A. and G. Abowd (1996). Modelling status and event behaviour of interactive systems. *Software Engineering Journal*, **11**(6): 334-346.
- Dix, A. J. (1992). Pace and interaction. *Proceedings of HCI'92: People and Computers VII*, Cambridge University Press. pp. 193-207.
- Dix, A. J. (1995). Cooperation without (reliable) Communication: Interfaces for Mobile Applications. *Distributed Systems Engineering*, **2**(3): 171-181.
- Dix, A. J. and M. D. Harrison (1989). Interactive systems design and formal development are incompatible? *The Theory and Practice of Refinement*, Ed. J. McDermid. Butterworth Scientific. pp. 12-26.
- Fickas, S., G. Kortuem, and Z. Segall. "Software Issues in Wearable Computing." *Proc. CHI Workshop on Research Issues in Wearable Computers*, Atlanta, GA, U.S.,
- Fitzpatrick, G., et al, Physical Spaces, Virtual Places and Social Worlds: A study of work in the virtual, *Proc. CSCW'96*, ACM Press
- Gaver W., The Affordances of Media Spaces for Collaboration, *Proc. CSCW'92*, 1992, ACM Press.
- Gram, C. and G. Cockton, Eds. (1996). *Design Principles for Interactive Software*. UK, Chapman and Hall.
- Greenberg S., Marwood D., 'Real Time Groupware as a Distributed System; Concurrency Control and its effect on the Interface' *Proceedings of CSCW'94*, North Carolina, Oct 22-26, 1994, ACM Press.
- Henderson, A.J., and Card, S.A., Rooms: The Use of Multiple Virtual Workspaces to Reduce Space Contention, *ACM Transactions on Graphics*, Vol. 5, No. 3, July 1985.
- Howard, S. and J. Fabre, Eds. (1998). Temporal Aspects of Usability: The relevance of time to the development and use of human-computer systems – Special issue of *Interacting with Computers* (to appear).
- Hughes J., Rodden T., King V., Anderson K. 'The role of ethnography in interactive systems design', *ACM Interactions*, ACM Press, Vol II, no. 2, 56-65, 1995.
- Ishii, H., and Ullmer, B. (1997). Tangible Bits: Towards Seamless Interfaces between People, Bits, and Atoms. *Proceedings of CHI '97*, ACM Press.
- Johnson, C. and P. Gray (1996). Workshop Report: Temporal Aspects of Usability (Glasgow, June 1995). *SIGCHI Bulletin*, **28**(2).
- Johnson, C. W. (1997). The impact of time and place on the operation of mobile computing devices. *Proceedings of HCI'97: People and Computers XII*, Bristol, UK, pp. 175-190.
- Joseph, A., A. deLspinasse, J. Tauber, D. Gifford, and M.F. Kaashoek. "Rover: A Toolkit for Mobile Information Access." *Proc. 15th ACM Symposium on Operating System Principles (SOSP)*, Copper Mountain Resort, Colorado, U.S., ACM Press, Vol. 29, Pages 156-171. 3-6 December 1995.
- Lewis (1995). *The Art and Science of Smalltalk*. Prentice Hall.
- Long, S., R. Kooper, G.D. Abowd, and C.G. Atkeson. "Rapid Prototyping of Mobile Context-Aware Applications: The Cyberguide Case Study." *Proc. 2nd ACM International Conference on Mobile Computing (MOBICOM'96)*, Rye, New York, U.S., ACM Press,
- Patterson, J.F et al., Notification Servers for Synchronous Groupware, *Proc. CSCW'96*, ACM Press.
- Ramduny, D. and A. Dix (1997). Why, What, Where, When: Architectures for Co-operative work on the WWW. *Proceedings of HCI' 97* Bristol, UK, Springer. pp. 283-301.
- Ramduny, D., A. Dix and T. Rodden (1998). Getting to Know: the design space for notification servers. *submitted to CSCW'98*,
- Root, R.W., Design of a Multi-Media Vehicle for Social Browsing, *Proc. CSCW'88*, Portland, Oregon, Spetember 26-28 1988, pp25-38
- Roseman, M, Greenberg, S, TeamRooms: Network Places for Collaboration, *Proc. CSCW'96*, ACM Press
- Weiser, M. (1991). The computer of the 21st century. *Scientific American*, **265**(3): 66-75.
- Weiser, M. (1993). Some computer science issues in ubiquitous computing. *Communications of the ACM*, **36**(7): 75-84.

Wood, A., A. K. Dey and G. D. Abowd (1997). CyberDesk: Automated Integration of Desktop and Network Services. *Proceedings of the 1997 conference on Human Factors in Computing Systems, CHI '97*, pp. 552–553.