Interfacing with the Invisible Computer

Kasim Rehman, Frank Stajano, George Coulouris

Laboratory for Communications Engineering
Cambridge University Engineering Department
Trumpington Street
Cambridge CB2 1PZ, United Kingdom
{kr241, fms27, gfc22}@cam.ac.uk

ABSTRACT

The Ubicomp scenario of wirelessly networked processors embedded in everyday objects has been dubbed "the invisible computer". Users no longer interact with a computer but with familiar objects whose functionality is transparently enhanced by computing features.

Using the results of an extensive survey of past and current Ubicomp research (Rehman 2001), we highlight the major problem of this new style of interaction: because the computer is invisible, the user lacks an appropriate cognitive model for it, and cannot predict the behaviour or even the available features of the system. We argue that effective and usable Ubicomp systems will have to make the invisible computer visible.

Keywords

Ubiquitous Computing, Human-Computer Interaction, Visualisation

INTRODUCTION

Since the end of the first phase of Ubiquitous Computing (Ubicomp) which was marked by the deployment of initial prototypes, there have been a number of increasingly sceptical views on Mark Weiser's vision of hundreds of information appliances connected together in order to relieve the human being of common frustrations of traditional computing. These concerns are mainly associated with "over-automation" and the loss of control arising from it, the lack of appropriate feedback, the breakdown of traditional mental models for the system functionality and difficulties in using ubiquitous interfaces. We believe that most of these difficulties could be solved by concentrating on a simple idea that was originally part of the 'ubiquitous vision', but seems to have gone missing in recent research. "Many, many displays" was a point Mark Weiser made an effort to stress in his talk at UIST'94 (Weiser 1994). In the coming sections we will demonstrate how involving the user and presenting information to her in an appropriate manner can avoid new frustrations of the "third wave of computing".

NordiCHI 10/02 Århus, Denmark

THE PROBLEM WITH UBIQUITOUS COMPUTING

We will base our problem analysis on Norman's design principles (1990) and look at typical violations of these in Ubicomp.

Lack of a good conceptual model

Ideally such a model should be provided by showing affordances, constraints and mappings. Many of the new interaction styles we encounter in Ubicomp, however, do not permit this. Gesture recognition interfaces, for example, draw their naturalness from the fact that they do not need to present choices (affordances) the user would need to select. Tangible Interfaces (Ishii & Ull mer 1997) usually suffer from the fact that it is impossible to achieve a perfect mapping between the constraints of the physical object being manipulated and constraints in the virtual world. Sometimes the reason for missing affordances is partly of an aesthetic nature. One of the advantages of RF-id tags (Want et al. 1999) is that they make it possible to augment everyday objects completely unobtrusively, triggering some actions in the virtual world when brought close to a RF reader. The flipside is that we cannot really tell what will happen if we "use" the object or even which objects we can "use".

Loss of Control

An important class of Ubicomp applications are proactive triggering applications, such as a museum guide that automatically brings up the correct artifact description, or headphones that automatically associate themselves with a running CD player when brought close. The problem is that most of these applications do not provide an "override" facility. This is mostly not done due to feasibility (s hould headphones have an interface?) or the desire to hide the computer (most of the attraction of the electronic museum guide is due to the fact that pages come up by "magic").

Lack of Feedback

One of the applications we surveyed, for example, automatically puts your job at the front of the queue as you approach a printer. The question that remains to be asked is, what happens if your job does not come out right away? The user has to validate different hypotheses as to why the

system did not work as expected, since the standard printer user interface does not cater for these kinds of messages. Partly the problem is, that tasks in Ubicomp are often unstructured which makes it difficult to work out *when* and *where* to provide feedback.

Breakdown of traditional mental models

Ideally, interfaces should not only convey information about how to use them, but in their function as a surface representation (Norman 1993) also convey an image of the underlying system. We use the term *mental model* in order to distinguish the model a user builds of a system from the model the user builds about its usage (described earlier). The problem here is that in a world of distributed, invisibly interconnected computers, the "surface" no longer exists. Various researchers (e.g. Brown 1998) have reported that users of location-aware tourist guides got confused because the places being described, for example, were at least ten minutes walk away. The reason was the inaccuracy of the GPS sensors, which had resulted in the wrong page coming up. A more graphic description of this problem was provided by Beverly Harrison (1997):

[...] while I was at University of Toronto, we did some early work on what later became known as the "reactive room". Initially versions of the conference room were known as the "possessed room". Lights turned on and off when you moved or ceased moving. Equipment "spoke to you" through an electronic voice and told you that you had 5 seconds to "take action" or it would power down. Video signals switched to channels and "forgot" how to switch back.

A SOLUTION: THE PRINCIPLE OF VISIBILITY

If we wanted to crystallise the problem in one sentence we may say the following: Ubicomp systems try to stay out of the user's sight, but more importantly out of her mind, whereas a well-designed interactive system does make itself noticeable at some points. By now it should be clear why we believe that Ubicomp systems should be *interactive*. The question is how we can adapt traditional interactive system design to Ubicomp systems, considering that task or machine may not be structured or well defined respectively.

So, how can we design a user interface for a Ubicomp environment that

- maps affordances and constraints between the real world and the virtual one as naturally as possible
- can offer choices to the user
- can provide feedback anywhere at any time
- represents a uniform "surface" to a system that may consist of hundreds of interconnected devices

In our attempt to solve this design problems we have decided to consult yet another one of Norman's principles: the principle of visibility. We have come to the conclusion that an Augmented Reality system, that can dynamically place virtual information in the real world would best meet the requirements set out.

We shall first introduce Augmented Reality and then talk about our implementation so far.

AUGMENTED REALITY

In its widest sense any system that connects the real and virtual world can be labelled "Augmented Reality" (AR). As such, even tangible interfaces are examples of AR. The narrower definition involves a system that uses a head-mounted display (HMD) and a tracker (Feiner et al. 1993). The tracker continuously measures the position and orientation of the head to some real object and displays a 3D graphics on a see-through HMD that makes the virtual object appear to be placed at a fixed location in the physical world. Achieving a good overlay, also called registration, is notoriously difficult. The four existing tracking technologies are: inertial, ultrasonic, electromagnetic and vision-based.

The HCI-oriented reader may be a bit concerned about using such a technology for an interface. Our own scepticisms were overcome by considering that HMDs that are indistinguishable from ordinary glasses are already beginning to appear. Apart from that, various "visionaries" of computer science have predicted that the HMDs will eventually be used by a large part of computer users (Feiner in Jacob 1997, Dertousoz 1997). Even sceptics like Donald Norman (2001) believe that the "augmented human being" is inevitable (in fact he even goes further).

SYSTEM IMPLEMENTATION

The first decision was to choose the type of tracking technology. We decided upon a vision-based technology, not only because of its accuracy but also because of its low cost. Up until a few years ago developing a tracker used to take a considerable amount of effort and expertise. Recent years have seen a rise in the availability of ready-made toolkits such as the ARToolkit, developed at the University of Washington (Kato & Billinghurst 1999).

The ARToolkit can be trained with square markers that are typically smaller than the size of a hand. A cheap camera mounted on the user's head can recognise its position and orientation and use this information to render objects on the HMD (Sony Glasstron PLM-S700) in OpenGL, thereby creating the illusion of an object being placed right on top of the marker.

We were impressed by its static registration performance, i.e. when the marker's position relative to the head is not changing, one cannot notice any offset between the virtual object and marker. Dynamic registration, on the other hand, refers to overlay accuracy in wake of moving targets. Initially, we noticed a lag between updates of the graphics scene. A more powerful graphics card (GeForce3) solved this problem. The frame rate was found to be 29 frames/sec for overlaying elementary graphics.

Ultimately, our aim is not just to place objects in the real world, but to open up a rich information channel between the user and the Ubicomp environment. This includes not only the sophisticated encoding of virtual affordances and constraints in a manner that the user can make sense of, but also the provision of facilities to control the environment.

We will describe which architecture and interaction style could accomplish the second part in the next section. In order to quickly prototype sophisticated visualisations we have replaced the original OpenGL rendering with a powerful high-level graphics API (Open Inventor).

To generalise the system even further we have inverted the tracking system, meaning that we do not calculate locations of markers anymore but infer the head position and orientation from markers at known positions. We are deploying markers throughout the room with the aim of prototyping an office-wide tracking system. This is necessary because conventional AR applications involve only a small area, whereas our application has a comparatively wide operation radius, similar to the system by Newman et al. (2001). In order to do that we are storing marker positions in a database. Once finished we hope to have a tracking system similar to the Active Bats (Harter et al. 1999), at low cost and with higher orientation accuracy. Our aim is to be able to specify any 3D location and ask the tracking service to place information at that position.

In building our system we have encountered a number of usability problems. First of all, in order to be useful, the user has to calibrate her tracking system. This is required in order to calculate the offset between the camera mounted on the user's head and her virtual point of view in the graphics system. At the moment this requires about five minutes and is quite tedious. Considering that it needs to be done every time the HMD is moved slightly, it can become a serious drawback for a user interface. Secondly, we are considering separating the image acquisition from the image processing, which will make it possible to move around with a small mobile unit that can transmit the image to a server. Thirdly, the set-up, i.e. placing markers in the environment and measuring them out, does take some effort. We have devised an algorithm that can automatically add new markers to the database by inferring their positions from positions of known markers. Finally, we found that the discrepancies between the user's, the camera's and the HMD's fields of views lead to unpredictable system behaviour in the sense that virtual objects do not appear on markers that are visible. This, by the way, was another motivation to build a tracking service that does not depend on individual markers but on *any one* from a set of various markers.

In order to associate information with particular objects we can still attach markers to them and use marker recognition in order for the system to find out what object is located in front of the user. Once we have identified the object we can gain access to its resources and display its status information dynamically. We are researching into how to display what information in order to make the connection between real and virtual as intuitive as possible.

FURTHER WORK

Our long-term work includes the implementation of a reverse channel, i.e. from the user to the system. We have stated that our aim is to provide a uniform "surface" to the system. We believe that this necessarily involves abstracting from the heterogeneity of devices and data formats and going beyond the interface level into middleware.

Software Architecture

Let's take a simple example as an illustration: A user wants to switch all devices in a room to a busy status. What software architecture would be required to implement such a command?

Firstly, we would need a way of discovering devices. Secondly, we would need to have a way of describing a service or an appliance. Thirdly, we would need to have a way of sending commands to and receiving visual feedback from the services to the user's unit (that can perform some kind of filtering). Fourthly, we would need to provide services that can convert between different representations of data.

In order to meet these requirements we have evaluated various service discovery architectures. Right now, our plan is to use human-readable XML descriptions for services and use events to take care of messaging.

One of the aims, we wish to pursue with our user interface is to examine how we can encourage synergistic use of Ubicomp. Only by combining different appliances and services can the user unleash the full power of this new computing paradigm. In order to help the user achieve this we want to implement data conversion services in the middleware layer. Integration with our service description framework will let us show possible device interaction options visually by mapping semantic descriptions into visual ones. Some of our inspiration comes from the CyberDesk Project (Dey et. al 1998) and we are drawing on the resources of the QoSDREAM location-aware middleware project (Naguib & Coulouris 2001).

Interaction Style

We are not yet sure about which interaction style to use. Whether to select appliances in the real world or by their graphical representations, whether to use a graphical cursor or a tangible interface, or how to interact with virtual menus, are all unresolved issues at this point. The solution will probably involve multimodal interaction though the most obvious candidate is a speech interface, since it does not involve coordination difficulties for the user, is natural and does not require long navigation times.

Augmenting a speech interface with a HMD can possibly resolve some of the traditional difficulties of speech recognition by providing feedback. Also, since we are using a location-aware system, we can make use of the fact that limited vocabularies can be used at specific locations.

CONCLUSION

We have seen that a lot of current Ubicomp systems suffer from classical interaction problems. One possible reason is that qualifiers such as Invisible~ or Disappearing Computing, as Ubiquitous Computing has also been called, seem to imply that we should not see the computer at all.

But such an interpretation is fundamentally wrong. While it may be appropriate to hide the computing machinery, it is a mistake to deny its existence. The invisible computer is a complex system emerging from the synergy of many communicating and cooperating components. The user should be aware of the system, of its features, of the ways in which such features can be triggered, of the system's intended behaviour and of its current state. It is perfectly acceptable to hide the computer, but only if we provide in its stead an intuitive interface to observe and control the whole system.

For desktop systems, the GUI is understood and taken for granted. We still lack, however, a discipline equivalent to GUI design for screen-less ubiquitous computing. We need to analyse the new frustrations brought about by the invisible computer and develop a new set of principles for intuitive and ergonomic interaction with non-GUI systems.

Another lesson we can learn from the GUI: In pre-GUI times, we used to have a number of heterogeneous applications, all with their own formats and user interfaces. It was only with the advent of Windows systems that users could enjoy a unified experience.

Our approach is to involve, once again, computer graphics and visualisation in order to provide such a unified experience.

ACKNOWLEDGMENTS

The Cambridge University Board of Graduate Studies, the Cambridge European Trust and AT&T, Research Laboratories Cambridge sponsor the first author. Thanks to Andy Hopper and William Newman for comments on an earlier version of this paper.

REFERENCES

Brown P. J. 1998, Some Lessons for Location-aware Applications, pp. 58-63, *Proceedings First Workshop on HCI for mobile devices.*

Dertouzos M. 1997, *What will be.*, Piatkus, London.

Dey A. K., Abowd G. D., Wood A. 1998, Cyberdesk: A Framework for Providing Self-Integrating Context-Aware Services, pp. 47-54, *Proceedings IUI'98.*

Feiner S., Macintyre B., Seligmann D. 1993, Knowledge-based augmented reality, vol. 36, pp. 53-62, *Communications of the ACM.*

Harrison B. 1997, Position Paper for Ubiquitous Computing Workshop, *Proceedings Workshop on Ubiquitous Computing: The Impact of Future Interaction Paradigms and HCI Research at CHI '97.*

Harter A., Hopper A., Steggles P., Ward A., Webster P. 1999, The Anatomy of a Context-Aware Application, pp. 59-68, *Proceedings MobiCom '99.*

Ishii H., Ullmer B. 1997, Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms, *Proceedings CHI'97.*

Jacob R. J. K. 1997, UIST'007: Where will we be ten years from now?, pp. 115-118, *Proceedings UIST'97.*

Kato H., Billinghurst M. 1999, Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System, pp. 85-94, *Proceedings IWAR'99.*

Naguib H., Coulouris G.F. 2001, Location Information Mangement, *Proceedings Ubicomp 2001*, Available: http://www-lce.eng.cam.ac.uk/qosdream/

Newman J., Ingram D., Hopper A. 2001, Augmented Reality in a Wide-Area Sentient Environment, *Proceedings ISAR' 01*, Available: http://www.uk.research.att.com/abstracts.html#133

Norman D.A. 1990, *The Design of Everyday Things*, The MIT Press, Cambridge.

Norman D. 1993, *Things that make us smart*, pp. 79-90, Perseus Books, Cambridge, MA.

Norman D. 2001, *Cyborgs of the new Millenium*, Available: http://www.jnd.org/dn.mss/Cyborgs.html.

Rehman K. 2001, *101 Ubiquitous Computing Applications*, Available: http://www-lce.eng.cam.ac.uk/~kr241/html/101_ubicomp.html.

Want R., Fishkin K. P., Gujar A., Harrison B. L. 1999, Bridging Physical and Virtual Worlds with Electronic Tags, pp. 370-377, *Proceedings CHI '99.*

Weiser M. 1994, *Building Invisible Interfaces*, Presentation Slides, Available: http://nano.xerox.com/hypertext/weiser/UIST94_4up.ps