Proceedings of the Workshop

Pervasive Mobile Interaction Devices (PERMID 2005)

- Mobile Devices as Pervasive User Interfaces and Interaction Devices -

http://www.medien.ifi.lmu.de/permid2005/

in Conjunction with the

3rd International Conference on Pervasive Computing
(Pervasive 2005)

Enrico Rukzio, Jonna Hakkila, Mirjana Spasojevic,
Jani Mäntyjärvi, Nishkam Ravi (Eds.)

May 2005

Preface

Welcome to the Pervasive 2005 workshop Pervasive Mobile Interaction Devices (PERMID 2005) - Mobile Devices as Pervasive User Interfaces and Interaction Devices -. The workshop consists of 13 interesting papers which were reviewed by at least 2 PC members to help the authors to improve their papers for the camera ready version.

We would like to thank the authors for their contributions and the organizers of PERVASIVE 2005 for hosting PERMID 2005. Furthermore we would like to thank the Media Informatics Group at the University of Munich and the IST Project Simplicity funded by the EU for their support to publish this workshop record in a printed form.

We look forward to the workshop providing a rich environment for academia and industry to foster active collaboration in the development of pervasive mobile interaction devices.

Munich, May 11th 2005

Enrico Rukzio, Jonna Hakkila, Mirjana Spasojevic, Jani Mäntyjärvi, Nishkam Ravi

Theme of the Workshop

Mobile devices have become a pervasive part of our everyday live. People have mobile phones, smartphones and PDAs with them nearly always and everywhere. So far these mobile devices have been mostly used for phone calls, writing short messages and organizer functionalities. Currently we see that the development of context-aware services for mobile phones takes particularly the user, his situation, and the location into account.

But why not use these devices for interactions with the real world, as a mediator between the virtual and the user's world? While certain research domains within the fields of mobile applications and services advance at an amazing speed (e.g. context-aware services on mobile devices, using the sensors of mobile phones), the areas of pervasive mobile user interfaces, mobile devices as interaction devices, mobile devices for interactions with the physical world and user experiences in this field are still rather limited.

Goals

The main goal of the workshop is to develop an understanding of how mobile devices (particularly mobile phones, smartphones and PDAs) can be used as interaction devices. We will provide a forum to share information, results, and ideas on current research in this area. Furthermore we aim to develop new ideas on how mobile phones can be exploited for new forms of interaction with the environment. We will bring together researchers and practitioners who are concerned with design, development, and implementation of new applications and services using personal mobile devices as user interfaces.

Topics

Possible topics for the workshop include (but are not limited to):

- Interactions between mobile devices and the real world
- Augmented, virtual and mixed reality on mobile phones and PDAs
- Using mobile devices as user interfaces for terminals and vending machines
- Portable music players and personal servers as mobile interaction devices
- Multimodal interaction taking mobile devices into account
- Usage of sensors of mobile devices for pervasive applications
- Interaction metaphors for pervasive applications and services
- Gathering, management and usage of context information
- Interactive context-aware services on mobile devices
- User experience, user studies
- Applications and scenarios

Webpage

All information about the workshop, the papers and the proceedings are available at the website of the workshop **http://www.medien.ifi.lmu.de/permid2005/**.

Organizers

Enrico Rukzio
Enrico.Rukzio (at) ifi.lmu.de
Media Informatics Group, University of Munich

Jonna Hakkila
Jonna.Hakkila (at) nokia.com
Nokia Corporation

Mirjana Spasojevic
Mirjana.Spasojevic (at) hp.com
Hewlett Packard Labs

Jani Mäntyjärvi
Jani.Mantyjarvi (at) vtt.fi
VTT Technical Research Centre of Finland

Nishkam Ravi
nravi (at) paul.rutgers.edu
Rutgers University

Schedule and Table of Contents

A Mobile Device as User Interface for Wearable Applications

Pekka Iso-Ketola, Tapio Karinsalo, Manu Myry, Lasse Hahto, Hanna Karhu, Mikko Malmivaara and Jukka Vanhala

Tampere University of Technology - Kankaanpää Unit

Jämintie 14, FI-38700, Kankaanpää, Finland

Tel. +3582 5782 7142, fax +3582 5782 7141

(first name.last name)@tut.fi

ABSTRACT

In this paper, we describe the Tampere University of Technology's involvement in the research of machine washable and wirelessly rechargeable wearable technology and using mobile devices as input and output interface and as means of data transfer and processing for wearable applications.

Keywords

Wearable technology, smart clothing, wireless wearable system, machine washable electronics

1. INTRODUCTION

In the late 1990's many leading figures of wearable technology predicted that very soon wearables would overthrow the mobile handset and electronics would move in even closer to the user's body. The prediction is yet to come true and when the high-tech boom slowed down most became more cautious with their estimates. Wearable electronics haven't still made mass market, but teaming up with mobile devices could eventually solve some of the key problems in wearable applications.

The Tampere University of Technology (TUT) has conducted research in wearable technology since 1997 [1]. The TUT Kankaanpää Unit (KP1) was established in 2003 and its research concentrates on wearable technology and garment integrated electronics.

2. THE CHALLENGE

Current wearable applications still have several problems to be worked out before mass market. Many of them are related to the maintenance of the device, e.g. machine washing, recharging of batteries and customer service. Bringing a function close to the body can often be of more service than merely delivering hands free operating, but interfacing the wearable devices can be complicated. If the wearable device requires both input and output and needs to be operated on-the-go, an integrated wearable user interface may cause more problems than it solves.

First of all the usability may suffer if for example text input, or any multi-key input must be used; constructing a soft washable keypad is possible [2] but efficient typing may be difficult for lack of suitable rigid surfaces on the body against which to press the keys. If the keypad itself is made rigid the textile garment may no longer carry the added bulk, not to mention the obtrusive appearance. Secondly, for the same reasons high resolution display output is not yet an option as the flexible displays still really aren't flexible and tough enough to withstand regular garment wear [3]. Woven optical fiber displays [4] are softer in feel and lighter, but so far don't offer needed resolution or brightness. Thirdly, hard objects larger than a button or a zipper in soft textile are likely to damage the fabric in machine wash. Constructing them waterproof and rigid enough to be able to take a washing cycle is expensive and time-consuming. Finding a perfect location for the display and a keypad is hard from a usability and ergonomic point of view [5]. A shirt with a display can't show output if the user is wearing a jacket over the shirt. Even further, displays as well as most other output devices consume much energy in relation to sensor electronics and combined with wireless data transfer the overall energy consumption may require larger batteries that add weight and bulk.

A good alternative to integrated user interfaces in wearable technology would be to use a wirelessly connected mobile tool, preferably one with good input/output capability, one that is widely available and customizable for different target groups and tasks and one that nearly everyone is already familiar using and carrying along with them. Why make another wireless handset when most people already carry one wherever they go? By making the mobile phone a part of the system the toughest problems of added manufacturing (and purchasing) costs and the maintenance of the garment-integrated electronics could be solved.

3. THE SOLUTION

Figure 1. illustrates a wearable system linked with external devices with a mobile phone as the user interface and communication hub between the personal space and the environment.

Many of today's mobile handsets have inbuilt Bluetooth (BT) or even Wireless Local Area Network (WLAN), and support General Packet Radio System (GPRS) and can run third party applications. A garment with BT fits nicely into the wireless loop and can be accessed through a custom application fitted in a menu structure, which offers the actual service. BT may not be an optimal solution for implementing a Body Area Network (BAN), but, along with IrDa, currently the only standard system widely used in mobile phones. On the other hand BT is a more suitable protocol for a Personal Area Network (PAN), which requires a higher bandwidth.

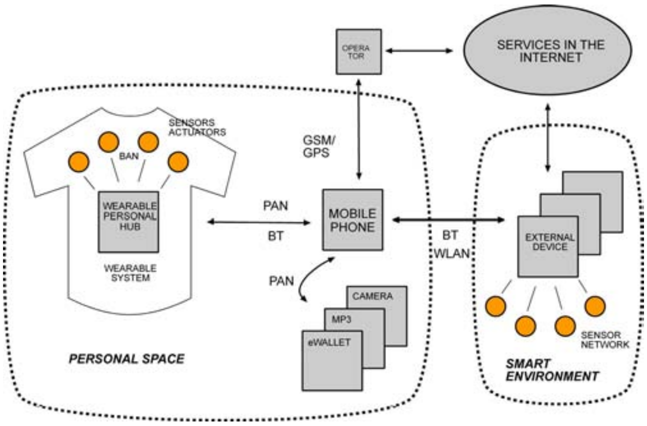


Figure 1. A mobile phone as the user interface and communications hub for wearable electronics.

3.1 Body Area Network

There are several options for creating a Body Area Network. They can be divided into three categories based on the data transferring method.

The first category is the most obvious and old-school, where the data between for example the CPU and the sensors is carried by wiring placed inside the garment. The wiring can be done by commercially available plastic-encased conductive textile yarns. The benefit of this method is that the wiring can include power supply for the sensors. A downside is the added sewing work when assembling the wiring to the garment; also the placement of the wiring needs to be done carefully to avoid dragging and restricting movement. A more sophisticated version of this is to include the conductive traces in the textile in the knitting phase, but for now this is a much too costly method. The traces could also be printed in the textile, but it too can be costly with the print having to withstand stretching, abrasion, moisture and sweat.

The second category builds on using the body's skin surface for sending signals. This requires a direct skin contact so it is most practical in garments that touch the skin surface. In this alternative all pieces of electronics need to have their own power supply.

The third category is wireless data transfer. Its benefits include less need for adapting the garment to the added technology and possibility to use short-range data transfer with moderate power consumption. Bluetooth is a good media for data loads around 1 Mbps, but a sensor-BAN however only moves a few bytes of data at a time and most of the BT channel capacity would be left unused. BT also consumes unnecessarily much power for a BAN. A better option for wireless data transfer in a BAN would be Zigbee [6], which has a capacity of up to 250 Kbps. Other alternatives include Wireless USB (WUSB) or Bluetooth v.2.

3.2 Personal Area Network

The BAN concept includes all parts of the wearable system located in the garment (e.g. sensor electronics) but to further include user's additional mobile devices and applications (e.g. mobile phone, camera, eWallet) we need to expand the concept of communication space from BAN to PAN which includes everything in the user's close proximity (Personal Space in fig. 1.).

The mobile phone serves a role as the mediating user interface between the BAN and PAN for reading the output from the garment sensors and acting as the input device for configuring the wearable device's settings. The mobile phone can be used to process the garment data; alternatively the phone can send the data further to an external service for processing over either a BT or GPRS or WLAN. Equally, an external device can monitor the environment where the user is located (Smart Environment in fig. 1.) and send input to the wearable system.

4. THE RESEARCH

The TUT Kankaanpää Unit is strongly focused on the wearables-infrastructure and has experience in manufacturing machine washable electronics by casting flexible circuit boards (FCB) in soft polymer. Several rounds of casting experiments have been made with help from the polyurethane specialists at Pucast Oy of Vammala, Finland and the results have been very promising; the cast electronics have proven to be unobtrusive and non-restricting.

4.1 Polymer Casting Electronics

Two kinds of molds were built for casting: closed molds with a pouring channel and an air groove, and open molds with layered construction. The open casting process is smoother from a manufacturing point of view, and depending on the accuracy of the mold there's little if any need for finishing a hardened, ready cast (Fig. 2.). Also 3D-shapes can be made. Closed casting also requires an amount of pressure to fill the cavity evenly and any flexible items inside the mold such as the FCB or fabric may twist or bend and place unevenly in the cast.

Fig. 2. Open-cast machine-washable, flexible, polymer-cast circuit board. Note the microphone in the cavity.

4.2 Testing the Prototypes

We built and cast three FCBs meant for testing ability to withstand machine washing. The wash test-boards were tested one at a time in a regular household washing machine at 40˚C/104˚F with color detergent. The program was set to spin-dry at a normal 900rpm. When the two first washes produced the same result, the third FCB was spared for later.

The wire contact of both FCBs had apparently taken some knocks in the wash and a connection was achieved only when bending the board at the contact (Fig. 3.).

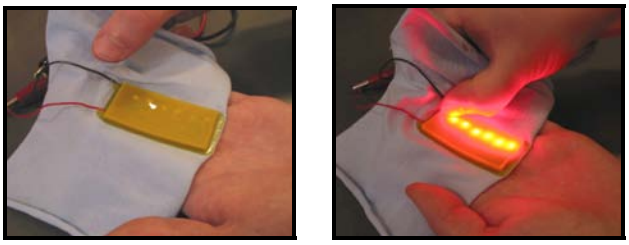


Fig. 3. The wash-test FBC was damaged at the contact.

The etched copper-tracing on the FCB was broken; the traces of the hand-made FCB were probably too thin to handle the strain put on by the mass of the soldered wire contact. The wires of the test boards were ordinary electric copper wire and the result would probably have been better with a lighter and thinner conductive textile wire. The tests proved the problem is mechanical, as the boards functioned right when bent at the corner. This can, on the other hand, be interpreted as a successful, water-proof cast.

The TUT is building a device for testing and monitoring flexibly encased electronics. As seen in the previous tests the ability to withstand mechanical strain is the main concern with FCBs. With the new system the connections within and between rigid and flexible materials can be tested. Also the layout design of FCBs will be evaluated as the placement of flexing points between more rigid areas can crucially affect the long-term reliability of flexibly encased electronics.

4.3 The Noise Shirt Concept

The TUT KP1 Wearable Electronics Maintenance Project set out to produce a functional prototype of a garment integrated electronic device implementing the research on flexible encasing and wireless recharging. We decided not to make the prototype an outerwear garment, as most current intelligent garment concepts tend to be outerwear due to their forgiving ability to carry bulky devices. Instead, we built a concept of a shirt, which would be easier to demo and which would clearly show how well we succeeded in making a wearable electronics shirt with permanently integrated devices.

We first and foremost wanted to test polymer-cast electronics, and LEDs were decided to be the best way of seeing if the device still works after different phases of testing. We also decided to make the actual function as simple as possible to be able to concentrate on the essential.

A concept of a Noise Shirt was accepted for a makeshift function. A microphone measures the surrounding environments noise level and shows it as a vertical 5 step ‘equalizer’ bar with the LEDs. Each LED marks a rise above a certain decibel level. The three lower LEDs are green and go from 65dB to 84dB. The top two LEDs are red and mark a noise level exceeding 85dB and 100dB (Fig. 4).

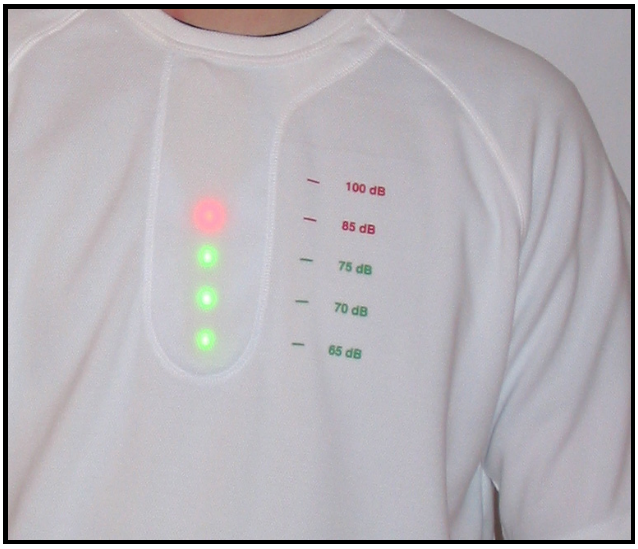


Figure 4. The Noise Shirt LED-panel.

Continuous exposure exceeding 85dB is the limit set by the EU authorities for recommended use of hearing protection. The device has a small battery with a wireless recharging induction loop in the neck tab. The garment is functional whenever charged and won't require any user input.

4.4 Wireless Recharging

Even after a wearable concept has been stripped of an integrated user interface it will still need power to operate the sensor logic and wireless data traffic. Our Noise Shirt concept of course needs a power source too. Water-proof connectors seemingly suitable for a smart garment concept are large and expensive. For usability reasons we also wanted to avoid a need to plug in the shirt after use or care. We knew that our concept wouldn't consume much power so wireless power transfer would be an alternative. After a little research inductive power transfer [7] seemed the best option for our purpose.

The most sensible and simple interface for wirelessly recharging a shirt with wearable electronics is a clothes hanger. As the two inductive coils need to be aligned as centrally as possible, a clothes hanger would automatically position them over one another. The hanger would ideally incorporate all the electronics and it would be connected to a mains socket. The hanger hook seems the most logical place to interface. No functional conductive hook has been made yet; so far the wiring runs to the shoulder tip of the hanger.

The system was first constructed in a regular hanger, which could house the electronics and was sturdy enough to carry a flap with a flat, spiral coil (Fig. 5.). After testing and some modifications a custom hanger mock-up was built from fashionable, transparent acrylic. The new hanger was shaped to make the shirt's neck tab drape over the hanger neck, to make the two coils meet. More custom hangers will be built in the future.

Figure 5. The first version of the recharging hanger.

Initial tests proved the hanger concept to be quite suitable for recharging the shirt. The spiral coils need to be researched further, though. The first prototypes were coiled by hand from copper wire, but the best alternative proved to be to etch an FCB with a spiral coil; thus the recharging electronics as well as the battery could all be fitted to one board (Fig. 6).

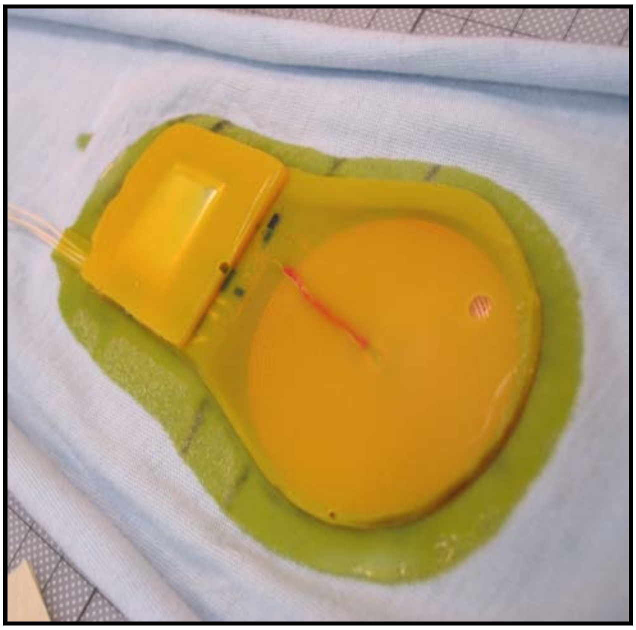


Figure 6. The neck-tab with recharging coil and electronics. A slightly messy cast.

The current system takes about 3 hours to recharge the empty Lithium-polymer battery and depending on the amount of ambient noise the Noise Shirt will run from 2 to 4 hours.

4.5 User Interface

The results in the field of infrastructure and maintenance solutions for wearable technology have shown that problems described in chapter 2 can be overcome with flexibly encased electronics and wirelessly rechargeable batteries. The next step in the research of the wearables-infrastructure is using wireless communications and a mobile user interface in order to access services and applications produced outside the garment.

The use of a mobile phone as an input and output device can solve interfacing problems associated with wearable technology and flexibly encased electronics. The input data to the garment is usually low bit rate information used just to control major functions (e.g. power control, operation mode). Text-form input is rarely used with sensory garments but as applications get more complex and new services are provided, the amount of input data needed to control the garment increases.

Most of the garment's output data is usually measurement data and in some applications text-form information on the garment's status. This output data can be either processed in the mobile phone to get the required results or passed on to a third party service provider for further analysis.

5. CONTINUING RESEARCH

The polymer-casting research goes on with the TUT Smart Polymer Project, and in the near future we'll get to cast in rubbers and different plastics. Another promising outlook is to get rid of the circuit board and print the traces on a fabric with conductive ink before casting. This would also require researching textiles as component base. The project could also venture into finding more ways of using cast polymers in wearable electronics, such as interference shielding or conductive polymer sensor patches for body monitoring. The recharging research goes on with customizing the spiral coils for each concept to make them as small as possible and building better custom hangers. Downsizing the recharging electronics continues alongside.

The TUT is also starting a new project on implementing wearable elements for well-being and fitness. For this project the use of a mobile phone as user interface is an important feature.

6. THE CONCLUSION

Everybody's still waiting for a wearables market to form and the killer-application to hit the shops, but with more effort and attention paid to the infrastructure around wearable applications both the consumers' and the manufacturers' confusion could be eased. So far the few commercial wearable products have been little more than marketing efforts, but when reliable and cost-efficient production technologies are developed, a commercial mass-producible application of wearable technology suddenly does not seem so far away. At least the problems in interfacing and maintenance may soon be solved, quite possibly by means of a mobile device interface and machine washable, fully garment-integrated electronics.

7. REFERENCES

[1] Rantanen, J., Impiö, J., Karinsalo, T., Malmivaara, M., Reho, A., Tasanen, M. & Vanhala, J., *Smart Clothing Prototype for the Arctic Environment*, Personal and Ubiquitous Computing, Volume 6, 1, p. 3-16, 2002.

[2] Post, E.R., Orth, M. *Smart Fabric or Wearable Clothing*, International Symposium on Wearable Computing, Proceedings, p. 167, 1997.

[3] Moore, S.K., *Just One Word –Plastics, Organic Semiconductors Could Put Cheap Circuits Everywhere and Make Flexible Displays a Reality*, IEEE Spectrum, September 2002, pp. 55-59.

[4] Gould, P. *Textiles Gain Intelligence*, Materials Today, October 2003, p. 38.

[5] Thomas, B., Grimmer, K., Zucco, J., Milanese, S, *Where Does the Mouse Go? An Investigation into the Placement of a Body-Attached TouchPad Mouse for Wearable Computers*, Personal and Ubiquitous Computing, Volume 6, 2, pp. 97-112, 2002.

[6] Evans-Pughe, C., *Bzzzz zzz [ZigBee wireless standard]*, IEE Review, March 2003, Volume 49, Issue 3, pp. 28- 31.

[7] Chan-Gyun, K., Dong-Hyun, S., Jong-Hu,P., Cho, B. H. *Design of a Contactless Battery Charger for Cellular Phone*, IEEE Transactions On Industrial Electronics, pp. 1238-1247, Volume 48, no. 6., December 2001.

Introducing Personal Operating Spaces for Ubiquitous Computing Environments

Anuroop Shahi
Intelligent Inhabited Environments Group & Chimera
University of Essex
Wivenhoe Park
Colchester, UK
ashahi@essex.ac.uk

Vic Callaghan
Intelligent Inhabited Environments Group
University of Essex
Wivenhoe Park
Colchester, UK
vic@essex.ac.uk

Michael Gardner
Chimera
University of Essex
Wivenhoe Park
Colchester, UK
mgardner@essex.ac.uk

ABSTRACT

Pervasive computing environments will combine everyday physical spaces with network aware devices and services; hence providing computing behaviour that is much more entwined with the environments we inhabit. Within such environments, a nomadic user will own and make use of many services attached to, and organised around, physical space. This study introduces the concept of a personal operating space: an entity formed for personalisation of space specific services, by treating a user's personal mobile device as an identity for personalisation.

1. INTRODUCTION

Pervasive computing causes us to examine ways in which existing computing infrastructures combine with everyday physical and environmental spaces; by understanding the dynamics of device rich environments; how devices are networked to correspond to the boundaries of physical space; and how users generally interact within and across these spaces: commonly referred to as 'smart space environments'. To coincide with the vision of pervasive computing, smart spaces will need to become part of the users background environment, and gradually become more ubiquitous in nature. These spaces will allow users to seamlessly access and use services across the myriad of devices provided by each space. Achieving this level of seamlessness requires true interoperability across heterogeneous devices, networks and applications. Much of this work is being lead by standards bodies such as the IEEE, OMG, W3C and the UPnP foundation, which all recommend their own standards for addressing interoperable components needed in smart space environments. Examples include bluetooth for personal area radio communication; TCP/IP for network communication; object and service orientated middle-ware technologies, such as CORBA and Web Services; and UPnP for communication between everyday devices in buildings. Additionally, methods for semantic interoperability are being realised by the introduction of semantic web technologies for terminology definition and mapping. Services can then use a shared ontology to develop methods for interoperation during spontaneous interactions (OWL-S). All these technology are well known for forming an integral part of any ubiquitous computing environment, with the challenge being to combine these to offer new types of behavior; characterized by being considerably more powerful and seamless than services today.

At the University of Essex, we are examining ways in which people interact with everyday spaces such as intelligent buildings. People tend to make use of different spaces over time, with each space differing depending on person, group or context (such as rooms in a building). People are also visiting foreign environments, such as buildings that offer a range of services from application based services, such as media applications, through to heating and light control. Furthermore, these nomads are increasingly carrying mobile devices, the most common being smart phones, which are essentially being treated as personal devices. The aim of this work is to treat these ubiquitous personal devices as a mechanism for interacting with and configuring, in a personalised manner, pervasive computing spaces; therefore creating a personal operating space between a user's personal devices and any services within the local space. A personal operating space will allow a user to both import a personal environment into the current space, along with personalising any services and devices provided by the space. This study seeks to build on theory from both nomadic computing [7] and ambient intelligence [9], and apply this in the context of ubiquitous computing; hence turning to invisibility and smart spaces as the main criterion for success.

2. SCENARIO

The following scenario should help crystallise the notion of a personal operating space.

2.1 Hotel Room

Bob arrives at his hotel room after a long tiresome journey. As Bob enters his room, a symbol on Bob's phone starts to flash in an unobtrusive manner. Bob now knows he is within a 'smart space', and decides to read his RSS based News Headlines. Using his phone, Bob selects the smart space menu, which has now become 'active' by the phone implicitly merging itself into the space. After an authentication procedure between Bob's smart phone and the smart space, Bob is presented with two menus: personal space and control space. Control space gives Bob the capability to 'control' his current environment (such as lighting and temperature etc), therefore using his personal phone as a universal remote control device. Personal space allows Bob to import his personal preferences into the current smart space, thus personalising the set of services offered by the space. Bob hits the personal space menu on his phone causing the smart space to present Bob with a set of application services available within the current space. Each application service is abstracted into 'tasks', such as 'Email', 'News', 'Music Streams', 'Clipboard' etc. Bob selects the 'News' menu, which causes the smart space to invoke an application that can handle RSS News feeds. When booting the application, the smart space configures the application to use Bob's preferences, thus retrieving Bob's personal selection of NEWS feeds and blogs. The application's display output is piped to a high resolution screen within the room. In the case of multiple screens being present within the space, Bob may simple choose to teleport the display to an alternative screen, which could be present within the sleeping area for instance. Again, with his smart phone as a remote control, Bob navigates over the various NEWS feeds.

Whilst reading his set of web feeds, Bob gets irritated with the temperature in the room. Instead of fiddling with the thermostat, Bob opens the 'Control Space' menu using his phone, and then clicks on the temperature menu. Using this standardised menu, Bob alters the room temperature using the joystick control on his phone.

After checking out of the hotel, Bob's personal agent confirms that all personal preferences have been removed from the visited smart space.

3. PERSONAL OPERATING SPACE

Currently, we are examining the concept of personalising space based media services by treating a user's smart phone as an identity, which is linked to a network profile holding a user's preferences: e.g. a list of RSS subscriptions. Our aim is to combine as much of this profile as is needed, into the user's current space by considering any constraints associated with the space, e.g. matching a user's preferences with a set of media services offered by a hotel room. We are also looking at infrastructures that allow mobile devices to seamlessly become part of spaces within intelligent buildings, and subsequently control any devices and services offered by a space. Figure 1 details a high level architecture, illustrating our personal operating space infrastructure, which allows mobile devices to combine with the local space, and invoke any services offered. Each component has been briefly described below:

3.1 Mobile Device Mediator

As mobile devices enter a particular space, the mobile device mediator (MDM) performs server beaconing via one of its sensors, therefore detecting any mobile devices within the current space. Our current prototype employs the now pervasive bluetooth technology for device detection and communication between a smart phone and MDM. Other wireless technologies may be used depending on granularity of a space. For example, one may wish to split a room into lots of mini -spaces by using sensing technology such as RF-ID. Alternatively, a space could span the whole building, therefore using WiFi technology. We believe bluetooth is the best of the current RF technologies for defining the boundary of a room based space; hence aligning with the theory of our behaviour being associated with the room that we are in, and thus so our control needs [9].

The main role of the MDM is to authenticate mobile devices appearing in the space, and mediate service events between mobile devices and any devices/services within the building. Once a mobile device has authenticated itself to the space, the MDM invokes the context model component to gather a list of services in the current space. This list is then translated into a form interpretable by the mobile device, and then transferred to the phone. Our current prototype uses a low-level feature associated with Sony Ericsson phones, for the installation of temporary hierarchical menus over bluetooth RFComm channels. We believe this approach demonstrates a key point in that users with SE phones need not install any software on their mobile devices to interact with a space. This essentially makes the whole process much more invisible. Embedding this same feature in the operating systems of other mobile devices, would essentially allow nomads to interact with smart space environments in a seamless manner.

Intelligent buildings will typically have one MDM per room, which depending on sensor granularity, could serve multiple spaces.

Figure 2 (going from left to right and top to bottom) shows the installation of services offered by the IIE space (intelligent inhabited environments room). As shown, the 'IIE space' has a menu for invoking the 'control space' of the room, together with various services such as 'NEWS' and 'Notice-board'. Once the control space menu is hit, the user is made aware of the fact that lighting may be controlled. Using the phone, the user may select the lighting menu, and subsequently select the 'switch on' menu. This will then fire an event to the MDM, which will pass the event to the personal operating space, causing event notification to a specific handler (a UPnP control point) and turning the lights on. After becoming aware of services within a space, a smart phone may issue various commands that are passed from the phone to the MDM, which then relays commands to the personal operating space.

3.2 Context Model

Services in a space are handled by the context model component, which is divided into the virtual services and space map sub-components:

The virtual services component provides semantic descriptions of services discovered within a building. Semantic descriptions allow composition of services into complex workflows, together with providing information about how these services may be presented to a space as 'Tasks': figure 3

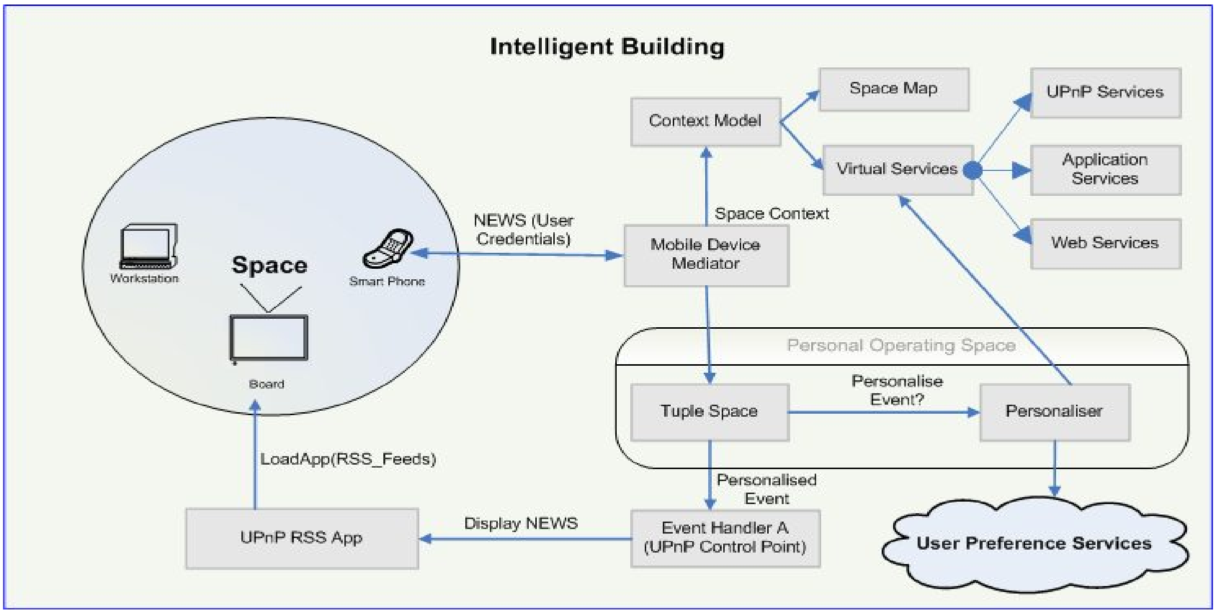


Figure 1: *Using a smart phone to interact with a space in an intelligent building*

shows an XML representation of services, which are used to generate a set of hierarchical interfaces for display on mobile devices - notice that this set of services was used for generating menus for figure 2. Studying the XML, 'menu' tags correspond to hierarchical menu based organisation of services; 'triggertask' tags correspond to actual events that are sent from a smart phone to MDM and then to the personal operating space. The XML 'task' attributes within 'triggertask' tags are used by the personal operating space to find a relevant event handler, e.g. the task 'http://essex.ac.uk/idorm#LightOn' will be passed to a UPnP device handler. We are currently using OWL-S to describe services. Depending on device capabilities, richer XML interface languages, such as XAML, may be used to present services to devices by transforming service descriptions (as in figure 2) to an appropriate display language.

The space map groups virtual services according to user defined notions of space. Here we intend to use techniques similar to Activity Zones [8], where building based regions are created from observed user behaviours such as movement, position and shape. Various techniques are then used to build a volume model of a person's trajectory in a space, using stereo computer vision. An activity map is then generated from this volume model. The map groups various sections of a physical space into areas of activity, which are updated with real-time information regarding users and other contextual information. Applications may then use the activity map to make decisions based on rich context models. A personal operating space will require a person specific-region centric view of space, to decide which devices and services to personalise, with regard to the device space surrounding a user.

3.3 Personal operating space and event handlers

The personal operating space is essentially divided into two components: the tuple space and the personaliser. The tuple space in this context is a network aware, event heap model [5] and allows for de-coupled, spontaneous and flexible interaction amongst services; making it ideal for nomadic interaction within smart spaces. Applications need not 'rendezvous' to communicate, but communicate indirectly by understanding the same event types; since reading, monitoring and restoring tuples can cause applications to perform appropriate actions. The personaliser component is used to personalise any services in the space. It works by using dynamic filtering processes to find relevant preferences for use with relevant virtual services.

Events from the MDM are sent to the personal operating space, which then attempts to personalise these events before dispatching them to an appropriate event handler. Once a suitable handler has been found, the event is passed to the chosen handler and processed accordingly. Event handlers express interest in events by subscribing to events published by the context model component.

4. RELATED WORK

Over the last few years, much work has been conducted in pervasive computing to bridge the physical and virtual worlds. The Cool-town project [6] is one such example, where things are given web presence by assigning URLs to everyday objects. Infra-red beacons are used to seamlessly send URLs to mobile devices, which then invoke these URLs to request an appropriate web page. We believe this approach is very well suited to assigning 'web information' to 'physical' things. However, some 'services' in a ubiquitous computing environment will not necessarily be identifiable through a tagging system or a formed based web page.

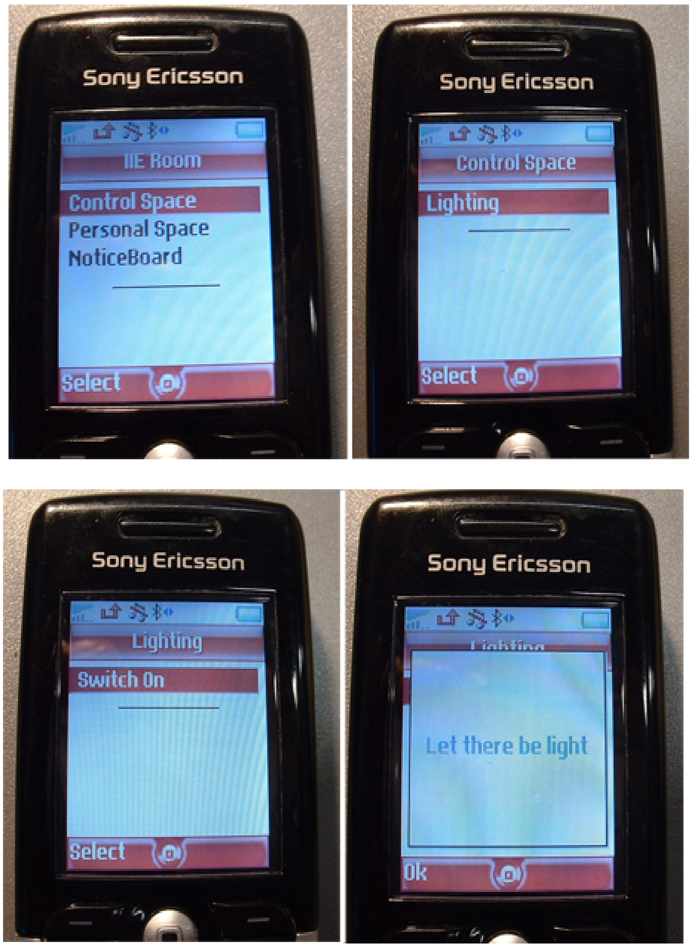


Figure 2: *Interacting with the IIE space using a smart phone*

```
<menu label="IIE Room">
    <menu label = "Control Space">
        <menu label = "Lighting">
            <triggertask label="Switch On" task="http://essex.ac.uk/idorm#LightOn"
            oncomplete="Let there be light"/> </menu>
    </menu>

    <menu label = "Personal Space">
        <triggertask label="News" task="http://essex.ac.uk/idorm#NEWS"/> </menu>

    <menu label = "NoticeBoard">
        <triggertask label="Add Note" task="http://essex.ac.uk/idorm#ADDNOTE"
        oncomplete="Enter NOTE on Board"/> </menu>
</menu>
```

Figure 3: *Describing a list of services for presentation to a mobile device in the IIE space*

For example, many intelligent building environments include services made available through middle-ware such as UPnP, Jini and other service discovery frameworks. Cool-town fails to address this by suggesting that web-pages be provided to interact with any service within a space. Doing this fails to address the benefits that certain, pervasive computing middle-ware technologies provide [4]. Our approach aims to support any type of pervasive middle-ware technology, by wrapping existing environmental services into virtual services using semantic web techniques. This way, the structure of the web is still utilised without removing support for any service specific middle-ware technology.

In terms of interaction, the context model represents space specific services in a hierarchical, task-driven manner. Recent studies have shown that a majority of mobile phone users now group and organise functions in a hierarchical manner [2]. This is quite different from the Cool-town URL approach, where URLs are displayed on a PDA with no contextual organisation. Our current MDM works by sending a hierarchical representation of services (figure 3) within a bluetooth defined space, for display to a mobile phone device. The MDM component can handle asynchronous events, therefore allowing environmental services to send various alerts/notifications to a mobile phone device. Finally, interacting with things in a space may require more varied and coarser levels of granularity; rather than pointing to or touching things. We are particularly interested in using short-range omnidirectional beaconing, such as bluetooth, rather than unidirectional technologies as deployed in Cool-town. We think that both types of beaconing granularity will be complementary, and suited to specific scenarios in pervasive computing spaces.

Another nomadic system is the Meeting machine [1], which embraces both nomadic computing and infrastructure-rich interactive workspaces. Here, interactive workspaces allow information to be shared within meetings. The Cool-town system [6] is used to provide nomadic access to the interactive workspace device. A remote control device is used to load files into a shared space, the e-table. We believe our mobile device mediator could be used to establish and coordinate multiple connections from users' personal devices (phones are starting to store huge files) to space specific services. This would provide users with another interaction mechanism for interactive workspaces. Generally, the personal operating space is primarily concerned with personalising a nomad's current space, whereas the meeting machine is concerned with allowing nomads to seamlessly share information in a collaborative setting.

Overall, the big differentiator between a personal operating space (POS), and related work, is the ability to personalise information and device based services within a user's current space. For example, the POS could be combined to personalise existing nomadic infrastructures, such as cool-town, through the filtering of space specific URLs based on a user's profile.

5. DISCUSSION

Considering the scenarios and high level frameworks, we can determine that a personal operating space will consist of a 'Control Space' and a 'Personal Space'. The challenge therefore lies in realising both of these, together with their underlying support infrastructure. An account of each POS component, and its significance, has been summarised below:

- Personal Space: today, most personalised computing environments tend to be fixed to a certain space. For example, a work based computing environment is typically accessed from a particular computer or network at work. Although applications/protocols do exist for allowing remote access to resources; for the lay person, these require considerable computing knowledge therefore being deployed by a small minority of users such as systems administrators and tech savvy users (even then a hideous amount of configuration is required). A personal space is a logical entity, which does not necessarily reside on a personal device of any sort, but is present in the network everywhere. For example, a user may enter an environment, and 'seamlessly' summon his or her personal environment using an appropriate device. Environments could be presented depending on context or manually selected by the user. This study is concerned with using the now pervasive mobile phone as a way to convey 'identity' to import a user's environment into a space, hence transforming a space into a user's personal space.

- Control Space: our environments are becoming increasingly augmented with devices of various shapes and sizes. Controlling these devices, especially in densely populated device areas, can often be an overwhelming task - just ask lecturers about lighting and projector control in lecture theaters. Lecturers do however carry mobile phones, and are more than likely to be familiar with these devices and their respective interfaces. Control space is therefore concerned with controlling everyday environments such as rooms, using mobile devices that we are familiar with.

Here at the University of Essex, we are examining the concept of a personal operating space, by considering the use of smart phone devices for personalisation of end user services, together with control of devices within our UPnP based intelligent building [3]. We have defined sample architectures, and are currently building concept demonstrators for evaluation.

6. REFERENCES

[1] J.J. Barton, T. Hsieh, V. Vijayaraghavan, T. Shimizu, B. Johanson, and A. Fox. The meetingmachine: Interactive workspace support for nomadic users. In *Fifth IEEE Workshop on Mobile Computing Systems and Applications*, 2003.

[2] S. Bay. Mental models of a cellular phone menu, comparing older and younger users. In *Mobile HCI*, 2004.

[3] V. Callaghan, G. Clark, M. Colley, H. Hagras, JSY. Chin, and F Doctor. Inhabited intelligent environments. *BT Technology Journal*, 22(3):233–247, July 2004.

[4] S. Helal. Standards for service discovery and delivery. *IEEE Pervasive Computing*, pages 95–100, Jul-Sep 2003.

[5] B. Johanson and A. Fox. The event heap: A coordination infrastructure for interactive workspaces. In *WMCSA 02: Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications*, page 83, 2002.

[6] T. Kindberg and J. Barton. A web-based nomadic computing system. *Computer Networks*, 35(4):443–456, Mar 2001.

[7] L. Kleinrock. Nomadic computing - an opportunity. *ACM SIGCOMM, Computer Communication Review*, 25(1):36–40, January 1995.

[8] K. Koile, K. Tollmar, D. Demirdjian, H.E. Shrobe, and T. Darrell. Activity zones for context-aware computing. In *UbiComp*, pages 90–106, 2003.

[9] S. Sharples, V. Callaghan, and G.Clarke. A multi-agent architecture for intelligent building sensing and control. *International Sensor Review Journal*, 19(2), 1999.

Product Reviews in Mobile Decision Aid Systems

René T.A. Wietsma* and Francesco Ricci
eCommerce and Tourism Research Laboratory, ITC-irst
Via Solteri n° 38
38100 Trento (TN) Italy
{wietsma,ricci}@itc.it
http://ectrl.itc.it

ABSTRACT

Recommender systems provide decision aid and information filtering functions that have a great potential application in the mobile context. An aspect which has not been extensively exploited, in current recommender systems, are ways to better explain the recommendations, for instance, exploiting the opinion of users about the recommended products. In this paper we shall describe the foundations for a mobile product recommender system which incorporates both structured (supplier driven) product descriptions and more subjective product knowledge, provided by users reviews. We think this type of recommendation technology could be especially useful in the mobile context, where people must take decisions in a rather short period of time, with a limited availability of product information, and with limited device capabilities.

Keywords

Mobile decision tools, meta recommender systems, electronic word-of-mouth

1. INTRODUCTION

A recent article in the Wall Street Journal [15] points out some success factors of travel blogs, i.e., websites where people can post their travel experience in the form of a review. This article indicates that people look for different ways to obtain and share more objective information about tourist products. Where a popular travel broker may say: "This is great" or "Try this out", the personal reviews are expected to give a more nuanced view, supported by ratings and rich media data (photos, video, etc.)

Providing users with relevant recommendation information is a difficult task. In addition to the technical components, such as the user model representation and the algorithms to generate recommendations, based on the user model, one has to design an appropriate graphical interface. This must include the functions required to support the user in the decision process, and convince her about the appropriateness of the recommendations. In the mobile context this can be regarded as a challenging task. Users tend to limit the interaction with the system for many reasons: connection and data exchange costs, environmental disturbances (noise,light,etc.), parallel activities of the user (driving, travelling, etc.). Besides these external influences, the device itself brings additional constraints, such as, small computation capabilities and limited input and output modality.

Our research focuses on methodologies and techniques for improving the user acceptance of product recommendations and for explaining these recommendations in the mobile context. To achieve this, we have developed a new approach where both product descriptions and user reviews are incorporated. We believe this approach, exploiting the hidden knowledge inside the reviews, will bring to the user more confidence on the recommendations and better product understanding. The products considered by this mobile recommender system are hotels and tourism attractions.

A user reviews can be described as a subjective piece of non structured text describing the user's product knowledge, experiences and opinions, typically also summarized by a final product rating. This non structured content introduces a number of research challenges, regarding the effectively usage of these reviews. In fact, sophisticated filtering and search techniques must be exploited to find relevant knowledge in product reviews. User opinions and products reviews are not new in the web scenario. Many product advise guides can be found on the Internet offering basic functionalities to find products, their specification and the user reviews [1]. These sites can be classified as consumer opinion platforms, which incorporate 'word-of-mouth' principles, and facilitating the exchange of product opinions between non experts [9]. There are many different motivators for these web sites [6, 9], but the most relevant for our research are: advice seeking and user involvement.

The innovative aspect of our approach is the incorporation of reviews in a structured way into the recommendation process. This means that we regard reviews as an alternative source of recommendation information, in which one can find product information, product experience and product popularity from a user's perspective. To exploit this kind of information, we use information retrieval and recommendation technologies [19]. Traditional systems, like Epinion [4], have not offered smart ways to incorporate user reviews in the product recommendation yet.

This approach could be highly beneficial in the tourism context where the tourist experience is the travel main motivation [20]. Products, or better said tourism services, lacks the

*René is affiliated with both ITC-irst and the University of Twente, Enschede, The Netherlands.

feature of "try-before-buy" or "return in case the quality is below expectance". This implies that (online) buying tourist products involves a certain amount of risk, which could be lowered by providing the user with a better product description and recommendation explanation. Another aspect, is related to the fact that a user buys more an overall experience [20, 5], rather than a single product. This experience involves three phases: anticipation (before), consumption (during) and memory (after). Reviews could contribute to the first and last phase by providing a way to gather information of similar travellers or to process the user own travel experiences.

2. RECOMMENDER SYSTEMS

Recommender systems are applications exploited in eCommerce web sites to suggest products and provide to consumers information for facilitating the decision process [18]. These systems offer personalized recommendations by incorporating the user characteristics, in terms of a persistent or ephemeral user model, and product wishes in terms of user queries. These systems implicitly assume that the user wishes and constraints can be converted, by using appropriate recommendation algorithms, in product recommendations. The recommendation algorithms either use knowledge of the domain, e.g., in content- or knowledge-based filtering approaches, or data about user's past behavior, in social filtering approaches.

The content based approach, filters the products according to the features specified by the user in a query [11]. This approach is especially useful when the product can be described by an extensive set of features. The user can specify her wishes either explicitly by providing product attribute values or implicitly, for instance, criticizing products shown by the system. Social filtering or collaborative filtering approaches can be described as technologies that try to mechanize the 'word of mouth' idea [8]. Here the recommendations are based on product rates given by a group of similar tasted people (neighborhood). Current research is mostly facing scalability issues and early-rater problems [16].

In our laboratory we have developed two recommender systems to support the user in travel planning. For the pre and post travel planning process, we have developed Nutking, a web based recommender system. Nutking enables the selection of travel locations, activities and attractions and supports the bundling of a personalized travel plan [12]. For the on-tour support, we have developed a mobile application, MobyRek [13], that supports critique-based product recommendations. In a critique-based recommender system, at every recommendation cycle, the user is allowed to criticize (i.e., to judge) the system's recommendation result [14, 3]. Both systems do not integrate reviews and, until now, we haven't found any recommender system that integrates reviews in the way we propose.

3. REVIEWS IN MOBILE DECISION AID SYSTEMS

3.1 Objectives

The overall objective of this research project is to design a methodology for mobile recommender systems that incorporates different knowledge sources and offer better recommendations in this context. The knowledge sources could be structured or not. In the first case, we consider product catalogues, where products are described with feature vectors, that can be easily queried with standard query languages. In the second case, we consider product reviews repositories, that are generated by a community of users interacting with a web site. To extract knowledge from this kind of repositories we will use social-filtering algorithms.

This methodology includes the design of a recommendation architecture in which the above mentioned sources of knowledge can be integrated together exploiting the best filtering algorithm for each available knowledge source. Some additional objectives, which will be derived from an empirical study, are the tuning parameters of each filtering algorithm and the approach for aggregating the results of the various filtering algorithms.

We aim at incorporating the previously mentioned hybrid product recommendation approach in a mobile application that users could access to obtain recommendations in a logical and straightforward way. The application will supply the user with additional functionalities (rank, sort and search) to revise the recommendations and better fit her wishes. Partly, these design principles will be derived from a user behavior study.

Moreover, we aim at providing an improved explanation of the recommendation to the user. In fact, in the currently available systems, the user only receives the recommended product together with an overall score indicating the appropriateness of such recommendation [7]. This kind of explanation is quite rudimentary and may raise doubts and questions to the user. We think to improve this explanation by providing the relevant reviews of similar tasted users. Moreover we aim at increasing the product acceptance rate. In fact, in current systems, a user is only able to retrieve the information provided by the supplier. As discussed before, travel decision making must consider the risk the service will fail to satisfy the user. Hence, we want to minimize this risk, and we hypothesis that reviews can contribute to attain this goal [6].

3.2 The Methodology

To convert our ideas into design principles, we have done a user behavior study in a group of 29 students. In this study we investigated the usage of product descriptions and reviews in the booking process of two different types of products: hotels and attractions. It is worth mentioning some differences found in the initial product filtering for the two product types. Hotel filtering was, in general, based on specific product features whereas for attractions people were more biased by the reviews. Another interesting result is the correlation between the perceived usefulness of product reviews and previous experience with product reviews web sites. These people, with previous experience in such type of content, where also more interested in negative reviews.

The design principles derived from this user behavior study together with our own ideas were incorporated into a system design. Since, one of the most important goal is to incorporate different knowledge sources we have adopted an meta recommender system architecture [17]. A meta rec-

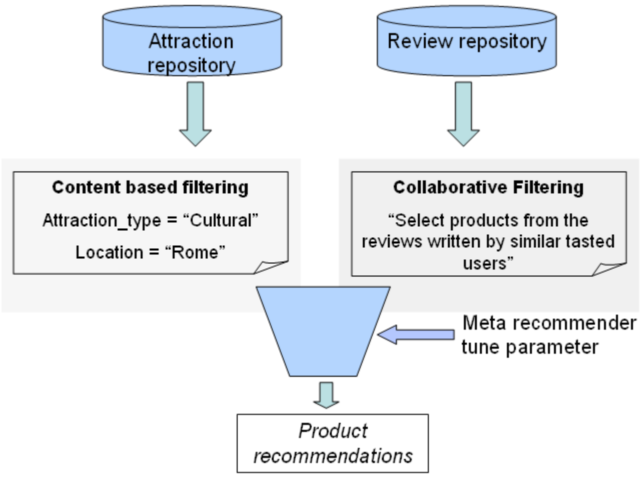


Figure 1: Meta recommender architecture

ommender provides users with personalized control over the generation of a single recommendation list, formed from a combination of rich data [17]. This personalized control can be implemented letting the user to explicitly control the influence of a specific data sources. Or, when using an implicit approach, it is up to the system, exploiting past user interaction, to decide how to balance the different knowledge sources. In our approach, if the product to be recommended has a rich structured description (e.g. a hotel), then the system tends to use more the content-based filtering approach. Conversely, if the product is poorly described, in term of attributes, then the system relies more on social-filtering. Burke [2] has also proposed to combine two recommenders into one 'mixed' hybrid recommender.

Figure 1 displays an overview of our approach. The attraction repository contains the tourist products to be recommended, and the review repository contains reviews about those attractions and a link to the review writer description. To filter the tourist product repository, we will use a content based filtering approach. The user can either specify the searched attraction or criticizes those recommended by the system.

Conversely, we will use a collaborative based approach to derive product review recommendations. We hypothesize that the products to be recommended are those having very popular reviews written by similar tasted users. Hereby we make a distinction between a positive review, which contributes in a positive way to the product score computation and a negative one that brings an opposite effect.

The final product recommendation score is computed by integrating the results of both recommenders. At this moment, we are considering different options regarding integration approach, such as a minimum or maximum function, which takes either the minimum or maximum score of both recommender as output, or a weighted average function, where the weights can be derived from either explicit or implicit user preferences for a certain recommender.

3.3 Screen Shot of the Prototype

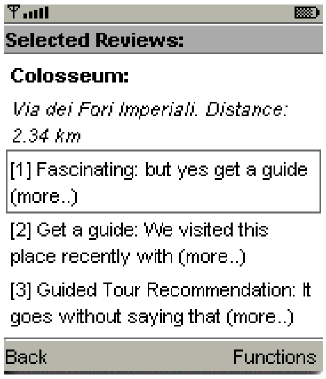


Figure 2: Screenshot of the prototype

Figure 2 presents a screenshot of the prototype recommender system. With this product recommendation, the user can find three summarized reviews written by the most similar users, that reviewed this product. In case the user is interested in reading the whole review, she can retrieve it by using the "Get complete review" functionality. Other functionalities are the retrieval of the next five best recommendations, the search for reviews satisfying some given constraints (text-length, date, review attitude) or those containing some keywords.

Before this stage, the user could filter some products and get recommendations, either entering her preferences or giving feedback, in the form of wishes and critiques, to some offered product recommendation.

3.4 Expected Results

3.4.1 *Increase Trust*

When booking a tourist product, the user wants to be sure that the product will satisfy her travel wishes and desires. In fact, nothing is worse than staying in a place you don't like. As we mentioned above, travel services cannot be "try-before-buy" and the traveller cannot return goods if they do not meet her taste. Our idea is that reviews will increase the user trust in the recommended products, since she can read more objective views, both positive and negative, of a large group of similar users.

3.4.2 *Increasing User Loyalty*

The availability of reviews can further push the user to share his knowledge and travel experiences. This makes the user more loyal to the system, since the user feels himself part of a community (system users). Moreover, the user can obtain recognition from other users, and this could eventually lead to a higher status inside this community [10].

3.4.3 *Better Product Understanding*

Reviews can be provided to the user as an explanatory tool. Reading comments written by similar people, the user can better understand if a product is (or is not) suitable for her. Product reviews can also be beneficial in understanding how to formulate a better product query and hence to simplify and shorten the human/computer interaction.

3.4.4 Exploring Hidden Knowledge

We hypothesize that reviews can provide 'local-available' knowledge. Local available knowledge can be described as knowledge you gain once you have visited the place, but you can not find using the product descriptions. For example, a review writer could suggest to visit a really nice fish restaurant after having enjoyed the view of the Garda lake.

4. CONCLUSION

This paper presents a new and innovative approach to incorporate different sources of knowledge in a mobile recommendation process, by using an meta-recommender approach. We believe that this approach will lead to more personalized and better recommendations.

Further research will be done in the field of user behavior modelling, exploiting a general agreed framework, such as that provided by an ontology. Besides this, we will look for ways to adapt the reviews to the mobile context. Hereby we will investigate summarization techniques and functionalities to retrieve reviews with a specific attitude, either positive or negative.

5. ACKNOWLEDGMENTS

René Wietsma is a visiting MSc student from the University of Twente, Enschede, The Netherlands, Chair of Human Machine Interaction (HMI). This paper is part of his MSc thesis work being done at ITC-irst.

This research work has been supported by the European Union, in the form of a Leonardo placement, contract number: "NL/04/A/PL/HO/123040" and by the Dr. L.A. Buma foundation, a foundation to support Frisian students, contract number: "05-10".

6. REFERENCES

[1] BBC. Net shopping guides lure surfers, url: http://news.bbc.co.uk/1/hi/technology/3736668.stm. 2004.

[2] R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.

[3] R. D. Burke, K. J. Hammond, and B. C. Young. The findme approach to assisted browsing. *IEEE Expert: Intelligent Systems and Their Applications*, 12(4):32–40, 1997.

[4] Epinion.com. Url: http://www.epinion.com.

[5] D. Fesenmaier, Y. Hwang, B. Pan, U. Gretzel, K. Grabler, A. Zins, and J. Mazanec. Tourist decision model. Technical report, DieToRecs IST-2000-29474, 2002.

[6] T. Hennig-Thurau, K. Gwinner, G. Walsh, and D. Gremler. Electronic word-of-mouth via consumer-opinion platformst. *Journal of Interactive Marketing, Vol.18, Nr.1*, 2004.

[7] J. Herlocker, J. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. In *In proceedings of ACM 2000 Conference on Computer Supported Cooperative Work*, 2000.

[8] J. Herlocker, J. Konstan, L. Terveen, and J. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004.

[9] Y. Jin, B. Bloch, and G. Cameron. A comparative study: Does the word-of-mouth communications and opinion leadership model fit epinions on the internet? Report, University of Missouri-Columbia.

[10] S. Kerner. Americans weigh in online rating: Rating is a popular pastime. *ClickZ Demographic Statiscs*, 2004.

[11] M. Montaner, B. L'opez, and J. Rosa. A taxonomy of recommender agents on the internet. In *Artificial Intelligence Review*, volume 19, pages 285–330. Kluwer Academic, 2003.

[12] F. Ricci, B. Arslan, N. Mirzadeh, and A. Venturini. Itr: a case-based travel advisory system. *Proceedings of the 6th European Conference on Case Based Reasoning*, 2002.

[13] F. Ricci, Q. Nguyen, and D. Cavada. On-tour interactive travel recommendations. *Proceedings of the 11th International Conference on Information and Communication Technologies in Travel and Tourism [ENTER 2004], Cairo, Egypt*, 2004.

[14] F. Ricci, Q. Nguyen, and D. Cavada. User preferences initialization and integration in critique based mobile recommender systems. *Proceedings of Workshop on Artificial Intelligence in Mobile Systems*, 2004.

[15] J. Saranow. Are travel blogs the big, new travel trend? *The Wall Street Journal*, 2004.

[16] Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of WWW10 Conference*, 2001.

[17] B. Schafer, J. Konstan, and J. Riedl. Meta-recommendation systems: user-controlled integration of diverse recommendations. In *Proceedings of the 11th conference on Information and knowledge management*, pages 43–51. ACM Press, 2002.

[18] J. B. Schafer, J. A. Konstan, and J. Riedl. E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, 2001.

[19] U. Shardanand and P. Maes. Social information filtering: Algorithms for automating 'word of mouth'. In *Conference on Human Factors in Computing Systems*, 1995.

[20] J. Swarbrooke and S. Horner. *Consumer Behaviour in Tourism.* Heinemann Educational Secondary Division, 1998.

Service Discovery in TinyObj: Strategies and Approaches

Pavel Poupyrev, Takahiro Sasao, Shunsuke Saruwatari, Hiroyuki Morikawa, Tomonori Aoyama

The Graduate School of Frontier Science, Tokyo University
7-3-1 Hongo, Bunkyo-ku,
Tokyo, 113-8656 Japan

{pavel, sasao, saru, mori, aoyama}@mlab.t.u-tokyo.ac.jp

Peter Davis

Adaptive Communications
Research Laboratories, ATR
2-2-2 Hikaridai, "Keihanna Science City"
Kyoto, 619-0288 Japan

davis@atr.jp

ABSTRACT
In this paper, we describe TinyObj service discovery model, which uses short wireless packet broadcasts for service discovery within a user's vicinity. We present a prototype implementation for the discovery model including the hardware and software development of a wireless discovery device. The developed wireless device, called Buoy, can be used as an independent service discovery device or as an attachment for cellular phones. The aim of the design is to provide the basic functionality for service discovery in ubiquitous environment. The developed software includes a uniform user interface that makes the system expandable and customizable. Indeed, a user can easily add, advertise, discover and remove any services in the system.

1. INTRODUCTION
Discovery of information is an essential problem in our daily lives. One of the easiest ways of finding information is using an Internet search engine that replies to user's queries with global search results. However, this way of information discovery provides little connectivity with the user's surrounding world since the search is conducted over a worldwide information database.

A new discipline, called ubiquitous computing, has been proposed [1-2]. Ubiquitous computing provides to a user a better connection with its surrounding world. In fact, this is achieved by embedding small computing devices in every object of the physical world. These devices can be equipped with short range wireless modules making the discovery of other devices and their services accessible to a user. We define this discovery as a real-world service discovery in proximity.

The real-world service discovery in proximity is different from the conventional ways of discovering services. Currently, a user discovers services upon request on the Internet or using other media such as newspapers. However, discovery in ubiquitous environment is performed in the background. Indeed, a user discovers services while moving from one place to another. This allows a user to find services in new locations or even newly advertised services in common visited places. The real-world discovery very well fits with the concept of ubiquitous computing where a user interacts with the real-world using background wireless computing devices.

A good example of a real-world service discovery in ubiquitous environment is the discovery of an ATM service. In this scenario, a user wants to find an ATM that belongs to a particular bank or provides VISA credit card operations. In order to do that a user has to define the ATM service discovery preferences specifying discovery details such as the bank name. When a user comes in proximity of an ATM, the user's device discovers the presence of the ATM. If a discovered ATM matches the specified preferences then the device alerts the user about its discovery.

The main contribution of this paper is the proposal of the *TinyObj model* for real-world service discovery in a ubiquitous environment and its *prototype implementation*. Since wireless ubiquitous devices have scarce resources, such as battery and processing power, we use a broadcast protocol eliminating the necessity for a complex routing protocol.

The prototype implementation has been developed including both software and hardware modules. We developed a wireless device for service discovery that can be used as an attachment with cellular phones or as an independent wireless discovery device. The developed discovery device is based on an original CSMA/CA-based MAC protocol.. The designed software provides to a user the ability of adding, advertising, discovering and removing new services using a uniform graphical user interface, unlike other discovery systems, which require knowledge of a programming language.

We will describe our prototype implementation covering three elements of the system:

Service matchmaking: The number of provided services usually greatly exceeds the number of services a user needs. The system should provide a mechanism for receiving focused search results. Also, service matchmaking should have a general way to represent any service. A user should be able to easily add and remove services from the system.

User Interaction: Our service discovery is meant to support two types of mobile users: a user with a cellular phone and a user with a wireless discovery device. One of the requirements is to provide easy user interaction with the device for both types of users.

Broadcast Protocol: Broadcasting is an important component of the system and should provide efficient data dissemination and consume as less power as possible. In this paper, we only discuss the requirements for the broadcast protocol.

The rest of the paper is organized as follows: In the next section we will present related works followed by an explanation about the TinyObj discovery model. In section 4 we will explain our implementation prototype, covering three major service discovery elements: service matchmaking, user interaction and network protocol approaches. In the last section we will conclude our work and describe directions for future work.

2. RELATED WORKS

A few systems, which support service discovery, based on short wireless communication, have been proposed such as nTAG [3], SpotMe [4], IntelliBadge [5], Conference Assistant [6], and Proxy Lady [7]. The goal of these systems is to empower users with additional features at conferences and meetings such as the discovery of friends, events or announcements, etc. Unlike these systems, which comprise pre-defined set of services, we target the development of extendable service discovery system where a new service can be easily added, advertised, discovered and removed.

Other research works, SLP [8] and JINI [9], provide a platform with an API for service discovery. However, implementation of service discovery requires knowledge of the programming language, such as Java or C/C++. In comparison, we focus on the development of a service discovery system that will provide a uniform graphical user interface as well as an API. Moreover, SLP and JINI systems are based on TCP/IP protocol, which requires computing devices with powerful processor, large memory and sufficient battery. Ubiquitous devices have strict limitations on available resources. Thus, it makes it more complicated to use these solutions.

Other research, such as ECA [10] and Smart-its [11], focuses on building hardware and software for ubiquitous computing. These projects provide a solution for quick implementation of new applications in ubiquitous environment where wireless devices are equipped with sensors and actuators to perceive context information. However, these projects do not particularly focus on service discovery issues in the ubiquitous computing.

3. DISCOVERY IN TINYOBJ

In this section we describe the TinyObj discovery model. We will use the same ATM service scenario to present TinyObj discovery. The TinyObj discovery model consists of four elements: 1) data initialization; 2) wireless data exchange 3) service matchmaking and 4) discovery alerts.

First, a user performs *data initialization*. A user can initialize two types of data: advertisements and preferences. For example, when a user wants to find an ATM service, a user specifies preferences (a filter) for service discovery as follows: only ATMs which support VISA credit card with a withdrawal fee of less than $1.5. In the meanwhile, a service provider, who advertises a service, may define a service advertisement like: Citibank ATM service, credit cards supported are VISA, MasterCard, and the withdrawal fee is $1.05.

Secondly, after the data has been initialized the *wireless data exchange* processes may begin. The wireless data exchange is based on sending periodic data broadcasts. It is not necessary for all participants to broadcast data.

Participants can be either active or passive. Active participants periodically broadcast data in proximity and passive ones save data on the wireless device without broadcasting. Normally, a service advertiser and a user can select whether they want to use active or passive discovery strategy.

In the ATM scenario, the bank uses an active strategy because it is interested in service delivery to the users. The users may instead use a passive strategy and store a filter on the device. The device tracks all services which match the stored filter.

However, it is possible to broadcast not only advertisements but also preferences. In this case, an advertiser receives preferences, which are compared with all advertisements stored in the device's database. Therefore, both a service advertiser and a discoverer can accomplish the discovery.

TinyObj concept is based only on broadcast transmissions. Thus, when a wireless device discovers a service it does not provide a method to reply to the sender using the same media. A user can access the sender via other media such as e-mail, phone, or web, using the contact information included in the packet.

Thirdly, after participants decide whether to be an active or passive discoverer, the *service matchmaking* process matches all received broadcast packets with the locally stored data. When a device receives an advertisement it is compared with all stored filters. Similarly, if a device receives preferences it is applied to advertisements stored on the device. The service matchmaking engine provides a function that takes an advertisement and preferences as parameters, and returns a comparison result in the form of a true or false value. If the returned result is true then a service matchmaking engine has succeeded in discovering.

Forth, after service matchmaking detects a match a device notifies a user with a *discovery alert* such as sound, vibration, etc. After viewing the content of the discovered service a user can access a sender using the contact information included in the data packet.

4. PROTOTYPE

Currently, the TinyObj system is implemented as a working prototype. We have built a wireless discovery device, called Buoy (Fig. 1). Also we have developed the device interaction software for cellular phones and personal computers.

4.1 Buoy

The Buoy device provides minimum functionality for service discovery. This includes storing/removing advertisements and preferences to/from the Buoy local database, performing service-matchmaking, communicating with a cellular phone or a PC, and broadcasting data.

Fig 1: Three variations of the wireless discovery device: the wireless device with an add-on component, a cellular phone and cradle

The discovery wireless device consists of two components namely a wireless module and an add-on component (Fig. 2). The wireless module includes the following components: a battery, an antenna, an Atmel ATmega128L processor, a Chipcon CC1000 wireless module, and two serial interfaces where one can be used by a cradle and another can be used by a cellular phone. The add-on component connects to the wireless module to provide a primitive input/output user interface equipped with a buzzer, two LEDs, a button and an Infrared port. The dimension of the device is 62x40x15mm.

4.2 Service Matchmaking

The service matchmaking is a vital component for successful discovery. The service matchmaking defines a format for service data and provides an algorithm for comparing advertisements and preferences. The format includes a list of name-value attributes to describe the service. This format enables the creation of services. However, our proposed system does not provide any access methods to the service but provides only service availability information. For example, if a user discovers a printer service, it may contain an IP address that can be used to reach the printer. The service matchmaking algorithm should provide narrowed search results because it is important to minimize the number of false alerts. For example, in an ATM service scenario, a user wants to discover ATM services satisfying some search preferences.

The system uses a service descriptor (Fig.3) that serves as a template to define advertisements and preferences. Also, the service descriptor includes information so that the TinyObj software knows how to represent data in a common way.

Fig. 3 shows an example of a service descriptor for an ATM service. The general parameters for a service descriptor are a service name, a description and a category. Attributes are optional parameters, which allow a user to provide more detailed information about a service. The attributes are useful for narrowing search results since they can specify search preferences more precisely. An attribute includes a name, a type, default values, a display and required options.

The type defines a data format used by the software to verify data input correctness and to represent the data. The type can be selected from the available types. The supported types in the current implementation are: Integer, Currency, Boolean, Select,

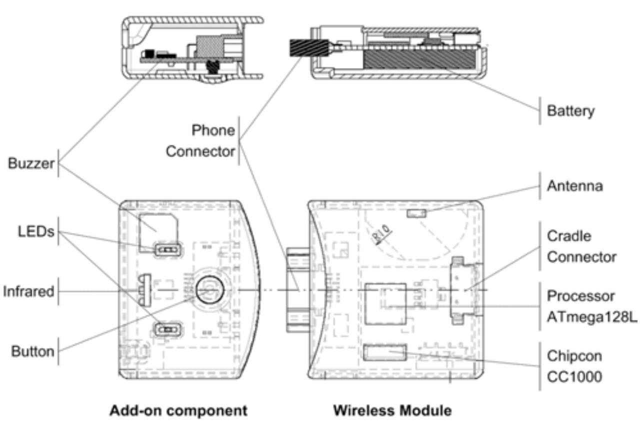


Fig 2: Buoy device schematics

Service name: ATM **Category:** Finance→Bank
Description: ATM service discovery

Attribute Name	Type	Default Value	Display	Required
Bank name	String	VISA, Master, Plus	☑	☑
Credit cards	Select		☐	☑
Availability	Select	Open, Closed	☑	☐
Charge fee	Currency		☑	☐
Location	String		☐	☐

Fig 3: A service descriptor example for an ATM service

and String types. The default value provides initial values when a new advertisement or filter is created.

A display option is used by the software for partial representation of advertisements. For example, Fig. 4a represents a list of discovered advertisements. Two ATM advertisements, Citibank and Mizuho, have partial representation depicting only three attributes: Bank name, Availability and Charge fee. In this case, TinyObj software represents only attributes, which have the display option enabled in the service descriptor.

Thus, each service has a service descriptor that serves as a template for the creation of advertisements or preferences. TinyObj software allows a user to create a new service descriptor using a graphical user interface specifying a service name, a category, a description and a list of attributes.

4.3 User Interactions

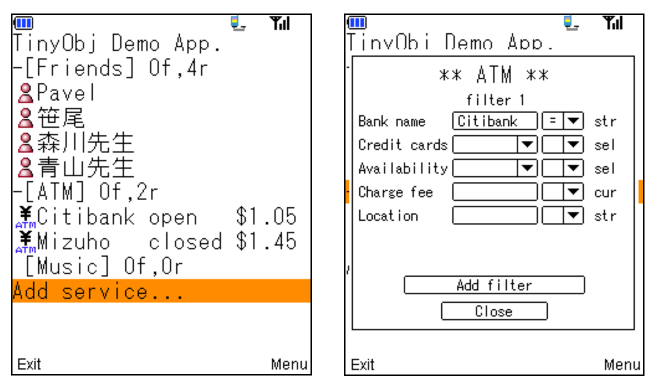
The TinyObj system is designed mainly for two types of users: a cellular phone user and a user that uses only the Buoy device without attaching it to a cellular phone. In this section we will describe interaction for both types of users.

4.3.1 Cellular phone user interaction

In Fig. 1, the Buoy wireless module is attached to a cellular phone. In order to advertise or discover a service, a user first has to store a service descriptor on the Buoy device so that the Buoy software knows how to handle a service.

In the current version of the TinyObj prototype, all created service descriptors are stored in the web-enabled database. A user can search for a necessary service descriptor using keyword searches or category searches.

After discovery of the service descriptor a user can create



a) b)

Fig 4: A cellular phone screen shots of TinyObj application a) a list of services registered with cellular phone b) an ATM service filter creation advertisements or preferences. When using service descriptors, the software can represent a form to assist the user in the creation of an advertisement or preferences. For example, Fig. 4b represents the case when a user creates a filter for discovering any ATM services that belongs to Citibank. After inputting the data, the user then specifies whether to choose an active or passive discovery mode. Finally, a user uploads and stores the input data and the service descriptor. Uploading a service descriptor is necessary if a user wants to create or edit the data later. Fig. 4a represents an example of stored service descriptors on the Buoy device including Friends, ATM and Music.

After all these steps, a user is able to start discovering. When a new service is discovered, the phone alerts the user through sound or vibration depending on the phone's settings.

4.3.2 Buoy device user interaction

A user who does not have a cellular phone is still able to discover services in proximity using only Buoy. However, the interaction requires an additional device, such as a PC or a cellular phone, only when initializing data or viewing discovered services. The actual discovery does not require the use of any extra device.

Buoy consists of two modules (Fig. 2): a wireless module and an add-on component. The add-on component has a primitive input/output interface with two LEDs, a button and a buzzer. Since the interface is very limited the use of an extra device becomes necessary in order to initialize the device as well as to view the discovered data. A user is provided with web-enabled PC software that can be accessed from the Internet. To initialize and view data with a computer, the Buoy uses the Infrared port..

Thus, the user interaction with Buoy works as follows. First, a user, using the web-enabled PC software, searches for a service descriptor. After finding the necessary service descriptor, a user creates an advertisement or a filter based on its content. Then, a user specifies whether to use active or passive discovery. Unlike cellular phone user interaction, a user selects an alert, such as flashing LEDs or a buzzer sound, for discovery notifications. Finally, a user uploads and stores these settings on Buoy using the Infrared port interface.

When Buoy discovers an advertisement, it notifies a user with the predefined alert associated with the service so that a user knows which type of service has been discovered. In order to view the discovered data a user has to use a desktop PC or a cellular phone.

4.4 Broadcast protocol

The TinyObj model is based on a broadcast packet exchange. Broadcast packets can be received by all neighboring nodes, which are in range of the transmitter. In our system the broadcast protocol does not provide a two-way communication scheme like usual routing protocols based on the query/reply model. This model has been selected to overcome the complexity and the limited scalability of typical packet routing protocol implementation especially in the case of highly dynamic mobile networks.

Broadcasting is a simple protocol that can be easily tuned to the application requirements. We define two requirements for our broadcast protocol, which are: 1) efficient packet dissemination and 2) minimization of power consumption. Efficient packet broadcasting is necessary because with the increase in the number of broadcasts the collision rate increases that leads to a poor packet delivery rate. The current prototype is based on CSMA/CA protocol that consumes large amount of power while listening to a channel. If we assume a constant current drainage of 110 mAh, the battery would last for approximately 6 hours.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we described TinyObj service discovery model, which uses short wireless packet broadcasts for service discovery within a user's vicinity. The broadcasting provides a simple way for data exchanges. We have presented a prototype implementation of the TinyObj model including both the software and the hardware. The hardware comprises the Buoy device that provides minimum functionality for service discovery. We also described user interactions with the Buoy device. The TinyObj software has been designed in order to provide a uniform graphical interface so that a user can easily add, remove, advertise and discover new services.

Future work would include studies on the implementation of power-efficient broadcast protocol and power-saving mechanisms. Moreover, the user interactions with the service discovery software should be improved to provide a generic platform. In order to do so, we are planning to conduct scenario field studies, adding new features to the system required by the different scenarios.

6. REFERENCES

[1] Mark Weiser, "Hot Topics: Ubiquitous Computing" *IEEE Computer*, October 1993.

[2] Mark Weiser, "Some Computer Science Problems in Ubiquitous Computing," *Communications of the ACM,* July 1993.

[3] http://www.ntag.com/, 2005

[4] http://www.spotme.ch/, 2005

[5] Donna Cox, Volodymyr Kindratenko, and David Pointer, "IntelliBadge™: Towards Providing Location-Aware Value-Added Services at Academic Conferences," in *UbiComp 2003,* Seattle, WA, USA.

[6] Anind K. Dey, Daniel Salber, Gregory D. Abowd and Masayasu Futakawa, "The Conference Assistant: Combining Context-Awareness with Wearable Computing," In *3rd International Symposium on Wearable Computers*, 1999.

[7] Per Dahlberg, Fredrik Ljungberg, and Johan Sanneblad, "Supporting Opportunistic Communication in Mobile Settings," in *CHI 2000 Extended Abstracts on Human Factors in Computing Systems*, 2000.

[8] Guttman E., Perkins C. and Kaplan S., "Service Location Protocol," RFC 2608

[9] Edwards W. K., "Core Jini," *Prentice-Hall*, 2001.

[10] Tsutomu Terada, Masahiko Tsukamoto, Keisuke Hayakawa, Tomoki Yoshihisa, Yasue Kishino, Atsushi Kashitani, and Shojiro Nishio, "Ubiquitous Chip: A Rule-Based I/O Control Device for Ubiquitous Computing," *Pervasive 2004,* pp. 238-253, 2004

[11] M. Beigl, and H. Gellersen, "Smart-its: An Embedded Platform for Smart Objects," *Smart Objects Conference (sOc),* 2003.

Camera Phones with Pen Input as Annotation Devices

Michael Rohs
Institute for Pervasive Computing
Department of Computer Science
ETH Zurich, Switzerland
rohs@inf.ethz.ch

Christof Roduner
Institute for Pervasive Computing
Department of Computer Science
ETH Zurich, Switzerland
rodunerc@inf.ethz.ch

ABSTRACT

This paper explores the use of camera phones with pen input as a platform for generating digital annotations to real-world objects. We analyze the client-side requirements for a general annotation system that is applicable in mobile as well as stationary settings. We outline ways to create and interact with digital annotations using the camera and pen-based input. Two prototypically implemented annotation techniques are presented. The first technique uses visual markers for digital annotations of individual items in printed photos. The second technique addresses the annotation of street signs and indication panels. It is based on image matching supported by interactively established 4-point correspondences.

1. INTRODUCTION

Digital annotations link user-generated digital media to physical objects. This allows users to combine the persistency and availability of physical objects with the flexibility and versatility of digital media. Physical media, like printed photographs and street signs, are tangible and permanent, but can typically store only a limited amount and type of information. Digital media, like text, graphics, audio, and video, are immaterial and volatile, but are virtually unlimited in terms of the amount and type of information they can represent. They can be automatically processed and shared across space and time. Using physical media as "entry points" [11] to digital annotations is a way to structure information and to embed it into the real world. In comparison to other types of augmentation the distinct feature of annotations is that users can freely create them and that they are not predefined by the system or any content provider.

Many projects have looked into annotating physical media with online information and services [2, 6, 7, 9, 11, 13, 14, 15]. Our goal in this paper is to explore the interaction possibilities of camera phones (or camera-equipped PDAs) with pen-based input as a platform for generating digital annotations. We present ideas of how to create and interact with annotations using phonecam-specific features. More generally, we are interested in how a mobile user interface for a generic annotation system could be structured that allows for the creation, access, sharing, and organization of digital annotations. This system shall be usable in stationary and mobile settings and exclusively rely on mobile devices as user interfaces.

Camera phones fulfill several essential requirements of a mobile annotation device. First, the camera in combination with image processing algorithms allows for identifying annotatable objects and for determining their orientation. There are multiple options for visually identifying physical objects, including image recognition techniques and visual marker detection. RFID tagging and near-field communication (NFC) for mobile devices [8] offer non-visual alternatives. Determining the orientation of objects in the camera image enables the registration of graphical overlays in the camera image in the sense of augmented reality [1, 4]. Second, wireless connectivity allows for sharing annotations with others, persistently storing and organizing them on a back-end server, and getting up-to-date information. Third, camera phones combine the ability to create annotations in multiple media types with the ability to play them back. Fourth, they are ubiquitously available in users' everyday situations.

A distinct feature of pen-based input devices is that they enable users to make more fine-grained annotations of objects captured with the camera. Users can encircle objects and create specific annotations to them, draw arrows to give directions, or put predefined icons onto the captured image. In addition to allowing for more fine-grained annotations, pen-input can also support image processing algorithms by telling the system which objects are relevant and which ones are not. In section 3 we show how this can be used to accurately segment street signs in images from the background.

Digital annotations can take many forms, such as text, graphics, audio, video, hyperlinks, vCard and vCalendar items, drawings and predefined graphical objects on the captured image. All of these media types can be created and presented on camera phones. If the semantics of the annotated object or its classification in a taxonomy are known to the system, then users might be provided with forms for rating objects or widgets for entering specific parameters. This supports users in minimizing the amount of data they have to enter into their mobile device in order to create an annotation. Annotations can further be supported by context data that is automatically gathered from the mobile phone [3], such as the current location or the time of day.

In section 2 we discuss the annotation of physical media with visual codes [10] in stationary settings. In section 3 we discuss the annotation of signs – or other areas with four clearly distinguishable corners – in outdoor environments, where attaching visual markers might not always be feasible.

2. DIGITAL ANNOTATIONS WITH VISUAL CODES

Even though digital photography has spread rapidly over the past few years, printed photographs are still omnipresent. To explore novel ways of attaching digital content to these artifacts, we have implemented a prototype application that allows for the annotation of pictures in a physical photo album (see Figure 1). We use a Windows Mobile 2003 based smartphone with an integrated camera to enable users to attach text or multimedia content (e.g. voice notes) to arbitrary parts of album pictures. In our approach, we stick a two-dimensional marker on every page of a conventional photo album. Each marker contains a unique number that we use to identify the album page. The markers can either be pre-printed onto pages or they can be supplied to users as individual stickers that they can put on album pages.

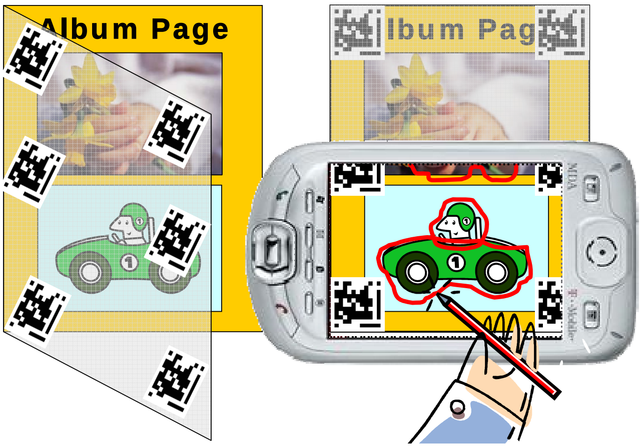


Figure 1: Digital annotation of a photo album with visual codes.

In order to annotate a photo on an album page, users take a picture of that page with their camera-equipped smartphone. Our annotation application running on the phone then extracts the marker from this snapshot and yields the numeric page identifier that it contains. At the same time, the snapshot is presented to the user on the phone's display. By drawing a polygon with the pen, users can then specify the part of the photo that they would like to annotate.

The phone then maps the polygon drawn on the display to a corresponding polygon in the physical photo's plane. In order to achieve this, we use the primitives available in the visual code system [10]. A visual code is a two-dimensional marker that provides a coordinate system which is independent of the camera's orientation. On top of that, the camera's distance, rotation angle, and amount of tilting relative to the code can be determined. These features allow us to transform the display coordinates of the user's drawing into a coordinate system in the physical marker's plane.

In our prototype application, users can attach plain text, hyperlinks, voice recordings, and files to a polygon by encircling an object of interest in a photograph. Along with the code's value identifying the album page and the polygon's coordinates in the code coordinate system, this annotation is sent over GPRS, WLAN, or Bluetooth to a back-end server, on which it is stored in a database.

Obtaining the annotations for a given album page works analogously: When the user takes a snapshot of an album page, the page identifier stored in the code tag is read. The application then fetches from the back-end service the coordinates of all polygons that are available for the given album page. These coordinates relative to the code tag are mapped to the corresponding pixels in the snapshot, which allows the application to highlight the polygons on the phone's display. Users can then read or play back the annotations for an object by tipping the polygon surrounding it.

The size of a printed code is currently 2x2 cm. The camera provides a resolution 640x480 pixels. With higher resolution cameras smaller codes (1x1 cm) can be used, which are less obtrusive. We experienced some difficulties regarding the placing of visual codes that we put on album pages. Depending on the size of an album page, the smartphone has to be placed relatively far away from it in order to take a snapshot of the whole page. As a result of this, the application occasionally could not recognize the visual code any more. We thus attached up to six visual codes to a single album page. This ensures that, when the camera is placed closer to the album and covers only a part of it, there is still at least one code located in this part. This, however, incurs the problem of an album part that, depending on how the camera is held, can be seen with a certain code first and another code later. Since each visual code has its own coordinate system, we needed a way to transform the coordinate systems into each other. This could be achieved by pre-printing the visual codes at fixed positions on album pages. In our prototype, we opted for another approach that still allows users to freely place stickers on a page. However, we introduced an additional step to initialize album pages before annotation. In this step, users have to take a few snapshots of a page that contain several codes at the same time. The application can thus learn about the arrangement of the markers and obtain the data needed to transform the coordinate systems of the different codes into each other.

The idea of annotated photographs was presented before in the Active Photos project [6]. However, the annotation and viewing process is different than with our prototype: The annotation of an Active Photo is done in a Web browser and relies on the availability of a digital version of the photo. Whereas we use a standard off-the-shelf smartphone, a special lap-mounted appliance is needed to interact with Active Photos. A third difference lies in the way regions with annotations are shown in a picture. While Active Photos are placed in a transparent envelope where marking objects offers additional content, we overlay the image shown on the smartphone's display with polygons.

Since our prototype builds upon the generic platform of smartphones and does not require the annotated object to be available in a digital version, it has a wide array of potential applications. It would be possible to annotate not just photo albums, stamp collections, or elements in a newspaper, but also all kinds of everyday objects ranging from product packages to posters and places in a city. Another field are applications where professionals such as the police or insurers need to annotate, for example, a crime scene, accident or damage. Architects could attach visual code stickers onto construction plans in order to add digital annotations (e.g. drawings with the stylus) while at the construction site. Yet another application area is medical diagnosis, in particular the annotation of printed X-ray images onto which visual code stickers could be pasted.

3. SIGN ANNOTATIONS WITH IMAGE MATCHING

Annotating objects by attaching visual markers is sometimes not an option, since the objects may not be under the control of the annotator, physically not reachable, or visual markers might be too obtrusive. This could be the case at public places, for example. Yet many objects, like street signs, shop signs, restaurant signs, indication panels, and even facades of buildings are sufficiently regular and have clear-cut borders to the background in order to be used as annotation anchors. Our idea for using signs as annotation anchors is based on interactive support by users and simple image matching, which makes the approach suited for execution on camera phones with pen-based input. Additionally, context data that is automatically gathered from the mobile phone is taken into account.

In order to attach an annotation to a sign or to retrieve annotations, users take a photo of the sign including any background with their camera phone. The result might be a picture as shown in Figure 2a. Users now tap the four corners of the sign on the device screen with their stylus (Figure 2b). A frame around the sign appears, whose corners have handles to allow for readjustment. This interactively supported sign selection approach solves two problems. First, if multiple candidate objects are present in the image, the one of interest to the user is selected. Second, the image segmentation process becomes trivial.

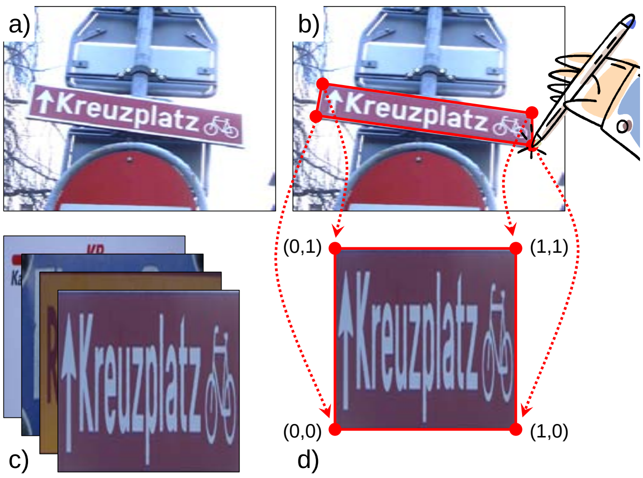


Figure 2: Annotating signs using camera phones with pen-based input: (a) captured photo, (b) framing a sign with the pen, (c) set of templates, (d) mapping framed area to unit square.

To enable simple matching of the framed part of the image (the sign) against a set of templates (Figure 2c), the framed part is projected into the unit square (Figure 2d). Depending on the orientation of the user towards the sign when taking the photo, the sign may appear perspectively distorted in the photo. This perspective distortion can be removed and the framed part projected into the unit square as follows. The four corners of the frame are set as correspondences to the corners of the unit square (Figure 2d). Since the frame corners are coplanar, there exists a unique homography (projective transformation matrix) between the pixel coordinates of the framed area and the unit square [5]. By scaling the unit square, we can thus produce a square request image of a predefined size (in our current implementation 480x480 pixels), which is sent to a server for matching against a set of template images. If the mobile device stores the relevant set of templates (Figure 2c), then the matching algorithm can also be run on the mobile device.

To further facilitate image matching and to make it more robust, we take a number of context parameters into account that are automatically gathered from the phone at the time of capture and sent to the server together with the request image. The context parameters comprise the current GSM cell id(s) for spatially restricting the search and the time of day (morning, noon, afternoon) to restrict matching to images taken under similar light conditions. The server may optionally add the current weather conditions (sunny, cloudy, rainy) to further restrict the search.

In our current implementation, the actual matching algorithm on the selected subset of templates is executed on a background server, which stores the shared annotations and templates. It computes the sum of differences between the request image and each template image by adding up pixel-by-pixel differences of the hue value. If saturation is below a certain threshold for a pixel, it adds the (gray) value difference. The server returns a list of matching annotations to the phone, ordered by increasing image difference values.

We expect that if we take contextual parameters, such as the current cell id, the time of day, and the current weather conditions into account, the remaining number of relevant templates will be a few dozen. Preliminary experiments on phonecam-generated images show promise that the matching algorithm can correctly distinguish that number of objects. A problem is of course that images of street signs are very similar – having a common text color and a common background color. Shop signs and restaurant signs typically show more variation in terms of color and visual appearance. Since street names are not unique, different physical signs with the same contents may appear multiple times along a street. In this case, annotations that refer to a particular location are not possible, but only annotations that refer to the street as a whole. This is true for all media that are reproduced multiple times – like flyers, product packages, images in newspapers, or logos.

The approach is beneficial for users, if it requires less effort to take a photo and tap the four corners than to enter some unique textual description of the annotated object (which of course needs to be identical across different persons if annotations are to be shared). Secondly, if the algorithm is not performed on the phone itself, the approach requires the upload of an image part and context data to the server via the phone network, which takes some time. We still need to investigate, how accurately users typically draw frames on a mobile phone with pen-based input and in what way imprecise frames degrade matching performance. In addition to simple pixel-to-pixel color comparisons, better image matching approaches need to be investigated [12]. For signs that have a clear visual border against the background it might suffice if users specify a single point on the sign. Image processing algorithms could then automatically extend the region based on color similarity and find the corners.

The presented approach is applicable if it is not desirable or impossible to attach visual markers to an object. Objects are recognized based on their unmodified visual appearance. Thus the facade of a building can be annotated even from a distance. A disadvantage is that a conscious effort is required for the user to retrieve annotations. Annotations are not discovered automatically, as is the goal in augmented reality systems [1, 4]. Still, there are a number of compelling applications, like pervasive gaming, in which the proposed sign annotation approach can be a component.

4. SUMMARY AND FUTURE WORK

We have investigated interaction possibilities that camera phones with pen-based input provide for creating digital annotations of physical objects. Camera phones fulfill the technical requirements of object identification and orientation detection, online connectivity for sharing annotations, the ability to create and play back annotations, and ubiquitous availability. Pen-based input allows for more fine-grained annotations. We have presented two digital annotation approaches that are applicable under different circumstances. The first one relies on visual code stickers and enables the annotation of individual items on a printed page. The second approach is based on the interactive establishment of a 4-point correspondence, which helps separating a selected area from the background and thus simplifies image matching.

We are going to investigate especially the second approach further, which is still at an early stage. Topics that need to be explored include user acceptance of tapping the corner points, the typical accuracy of the area framed by the user, better image matching techniques, a larger set of test images taken under different lighting conditions, as well as target applications that can be based on this approach. Applications that we intend to implement are restaurant recommenders as well as pervasive urban games that involve looking for sign annotations within a scavenger hunt. Another aspect is the creation of a coarse taxonomy of annotated objects that would allow for automatic processing of images and annotations. The background system could then automatically provide the user with other relevant shared annotations and related objects.

Acknowledgments

We thank Kaspar Baltzer for implementing the annotated photo album as part of his semester project.

5. REFERENCES

[1] R. T. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, August 1997.

[2] J. Barton, P. Goddi, and M. Spasojevic. Creating and experiencing ubimedia. HP Labs Technical Report HPL-2003-38, 2003.

[3] M. Davis, S. King, N. Good, and R. Sarvas. From context to content: Leveraging context to infer media metadata. In *MULTIMEDIA '04: Proc. 12th ACM conf. on Multimedia*, pages 188–195. ACM Press, 2004.

[4] S. K. Feiner. Augmented reality – a new way of seeing. *Scientific American*, 4 2002.

[5] P. S. Heckbert. Fundamentals of texture mapping and image warping. Master's Thesis, Department of Electrical Engineering and Computer Science, University of California, Berkeley, 1989.

[6] T. Kindberg, E. Tallyn, R. Rajani, and M. Spasojevic. Active photos. In *DIS '04: Proceedings of the 2004 conference on Designing interactive systems*, pages 337–340. ACM Press, 2004.

[7] P. Ljungstrand and L. E. Holmquist. WebStickers: Using physical objects as WWW bookmarks. In *CHI '99: CHI '99 extended abstracts on Human factors in computing systems*, pages 332–333. ACM Press, 1999.

[8] Nokia Corporation. Nokia Mobile RFID Kit and Nokia NFC (Near Field Communication). www.nokia.com/rfid and www.nokia.com/nfc.

[9] J. Rekimoto, Y. Ayatsuka, and K. Hayashi. Augment-able reality: Situated communication through physical and digital spaces. In *ISWC '98: Proceedings of the 2nd IEEE International Symposium on Wearable Computers*, pages 68–75, 1998.

[10] M. Rohs. Real-world interaction with camera-phones. In *2nd International Symposium on Ubiquitous Computing Systems (UCS 2004)*, pages 39–48, Tokyo, Japan, November 2004.

[11] M. Rohs and J. Bohn. Entry points into a smart campus environment – overview of the ETHOC system. In *ICDCSW '03: Proc. 23rd Intl. Conf. on Distributed Computing Systems*, pages 260–266, 2003.

[12] Y. Rui, T. S. Huang, and S.-F. Chang. Image retrieval: Current techniques, promising directions, and open issues. *Journal of Visual Communication and Image Representation*, 10(1):39–62, March 1999.

[13] M. Smith, D. Davenport, H. Hwa, and L. Mui. The annotated planet: A mobile platform for object and location annotation. In *1st Int. Workshop on Ubiquitous Systems for Supporting Social Interaction and Face-to-Face Communication in Public Spaces at UbiComp 2003*, October 2003.

[14] M. A. Smith, D. Davenport, H. Hwa, and T. Turner. Object auras: A mobile retail and product annotation system. In *EC '04: Proc. 5th ACM conf. on Electronic commerce*, pages 240–241. ACM Press, 2004.

[15] R. Want, K. P. Fishkin, A. Gujar, and B. L. Harrison. Bridging physical and virtual worlds with electronic tags. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 370–377. ACM Press, 1999.

Mobile Phones as Pointing Devices

Rafael Ballagas
Media Computing Group
RWTH Aachen University
Aachen, Germany
ballagas@cs.rwth-aachen.de

Michael Rohs
Institute for Pervasive Computing
ETH Zurich
Zurich, Switzerland
rohs@inf.ethz.ch

Jennifer G. Sheridan
Computing Department
Lancaster University
Lancaster, England
sheridaj@comp.lancs.ac.uk

ABSTRACT

This article outlines two techniques that allow the mobile phone to be used as a pointing device for public terminals and large public displays. Our research has produced two complimentary camera-based input techniques. We outline the details of the interaction techniques and identify further areas of exploration.

1. INTRODUCTION

Mobile phones are the first truly pervasive computer. They help us keep in touch with our loved ones, and help manage everyday lives with address book and calendar functionality; consequently, the mobile phone is always carried with us. Technological trends are packing more and more computational, communication, and sensing resources into the small, convenient form factor of today's smart phone. We leverage these trends to provide a ubiquitous pointing device using the mobile phone.

We have developed two complementary camera-based interaction techniques called *point & shoot* based on optical sensing and *sweep* based on optical-flow detection. Both techniques can be used for pointing, for example, to control a cursor on a large display. *Point & shoot* is used for absolute cursor positioning, while *sweep* realizes relative cursor movement. The techniques could be combined in the same task, depending for example on the distance to be covered or the pointing precision required.

2. RELATED WORK

Other systems have used personal devices for direct manipulation interactions with large displays and information terminals. The Remote Commander enables individuals to use a PDA to control the mouse and keyboard on a remote display using the PDA's touch sensitive display for mouse input and graffiti for text entry [6]. The C-Blink [5] system allows users to control a cursor on a large display using a mobile phone with a colored screen. The user runs a program on the phone that rapidly changes the hue of the phone screen and waves the phone screen in front of a camera mounted above a large display. The camera tracks the relative position of this signal to control the cursor on the display. Slifverberg et al. [8] have studied the use of the handheld joystick, increasingly more common on today's mobile phones, as a pointing device for information terminals. Madhavapeddy et al. [4] introduce techniques that use visual tags known as SpotCodes. Interaction involves using a phonecam to scan tags or to manipulate tagged GUI widgets. The main distinction of our design is that it can be used to select any arbitrary pixel, where Madhavapeddy's work only allows the user to select or manipulate tagged objects.

3. INTERACTION TECHNIQUES

3.1 Point & Shoot

The *point & shoot* interaction technique is illustrated in Figure 1. The user aims the mobile phone to the target on the large display. The contents of the large display appear on the phone screen, which acts as a view finder and is continously updated as the device moves. Aiming is facilitated by a cross-hair cursor in the center of the phone screen including a magnification of the area around the center. The magnified part is shown in the upper right corner of the phone screen. In *point & shoot* interaction, the user's locus of attention is on the phone screen. The cursor on the phone screen is active and the large display cursor remains inactive. *Point & shoot* is triggered by horizontally pushing and releasing the joystick button. As soon as the user "shoots," a grid of tags is shortly superimposed (approx. 0.5 seconds) over the large display contents, as can be seen in the middle part of Figure 1. The coordinate systems of the recognized elements are then used to compute the precise point on the large display that was targeted. Finally, a selection is issued on the large display at the target point.

The *point & shoot* technique utilizes Visual Codes [7] to determine absolute positioning information. Each code defines its own local coordinate system (shown in Figure 2), which is invariant to perspective distortion arising from the inherent mobility of the phone camera. This enables the mapping of arbitrary coordinates in the camera image to corresponding coordinates in the code plane (in our case, the large display). In the *point & shoot* technique, we use this feature to determine the precise absolute pixel coordinates of the point on the large display that corresponds to the user's cursor on the local phonecam display.

3.2 Sweep

The *sweep* technique utilizes optical-flow image processing, which involves rapidly sampling successive images from a camera phone and sequentially comparing them to determine relative motion in the (x, y, α) dimensions. This enables the camera to be used as a three degrees of freedom (DOF) input device. No Visual Code has to be present in the view of the camera, since the relative movement detection solely relies on the comparison of camera images. In our implementation, optical-flow processing is performed directly on the phone rather than on the computer driving the display. One advantage of this strategy is user scalability; the interaction technique easily scales to a high number of users. A disadvantage however, is the high latency (about 200 ms) with current hardware (a 104MHz ARM processor in the Nokia 6600 mobile phone) that occurs when calculating the (x, y, α) changes from successive images. Studies have shown that system lag has a multiplicative effect on Fitts' in-

Figure 1: ***Point & shoot*** **interaction: The left screenshot shows the phone screen as the user is aiming at the highlighted target. The cursor on the phone is active and the locus of attention and the cursor on the display is inactive. Pushing the joystick to the left indicates a selection at the location of the cursor on the phone display. Then the visual code grid is briefly displayed for computing the target coordinates as shown in the middle screenshot. The grid disappears again and a mouse selection occurs in the target region.**

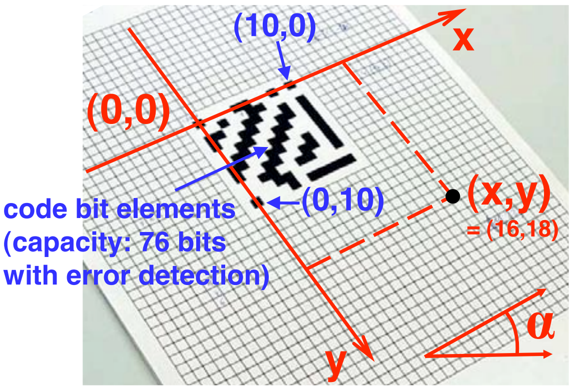


Figure 2: Each Visual Code has its own local coordinate system that is invariant to perspective distortion.

dex of difficulty [3], which is used to predict human performance in pointing and selection tasks. Yet mobile computing trends indicate that in the not too distant future mobile phones will have the processing power necessary to create a fluid interaction experience.

To invoke the *sweep* function, users vertically push and hold the joystick button, which acts as a clutch, to indicate to the system that they are actively controlling the cursor, and then they wave the phone in the air to control the (x, y, α) input. Users can release the clutch button to reposition their arm, which is similar to the way a mouse can be lifted to be repositioned on a desktop surface. This means that the camera need not be pointed directly at the display but can be pointed at the floor to allow users a more comfortable arm posture. In the *sweep* mode, the user can ignore the display on the phone and focus attention on the large display to observe the cursor movement.

3.3 Combining Techniques

To enable selection, dragging, and rotation, the *point & shoot* and *sweep* techniques are mapped to the phone's joystick button as shown in Figure 4. Absolute movement (*point & shoot*) is invoked by pushing the joystick in a horizontal direction. Pressing it to the left and releasing it again triggers absolute movement of the cursor only, whereas pressing and releasing it to the right also drags the object currently located beneath the cursor to the new cursor position. Relative movement (*sweep*) is invoked by pushing the joystick in a vertical direction. Holding it upwards invokes relative cursor movement only, whereas holding it downwards additionally drags the current object. Relative dragging includes rotation of the objects, which is done by rotating the phone around the z-axis. Absolute dragging includes rotation as well. Pressing the joystick key inwards (along the z-axis into the phone) is used for explicit selection. Mapping the interaction techniques to the joystick button in this way preserves simple one-handed operation and does not impinge on dexterity as the user is not required to reposition his or her finger to different buttons.

Figure 3: The sweep technique can be used to control a cursor by waving the phone in the air.

To more thoroughly illustrate the functionality of the combined techniques, we place our design into Card's design space of input devices [2] in Figure 5. The insight provided by the design space may help identify promising opportunities for new designs, and may form the basis of toolkits for composing individual designs. The three horizontally connected circles labeled *sweep* correspond to the 3 DOF and map to the (x, y, α) dimensions. Although it is possible to also detect relative Z movement and relative X and Y rotation, we exluded it here in order to focus on the most important data points. In our implementation, relative rotation around the X axis (dR:rX) is equivalent to linear Y motion and relative rotation

Figure 4: Phone input interaction: ***point & shoot*** **is mapped to horizontal joystick push-and-release,** ***sweep*** **is mapped to vertical push-and-hold.**

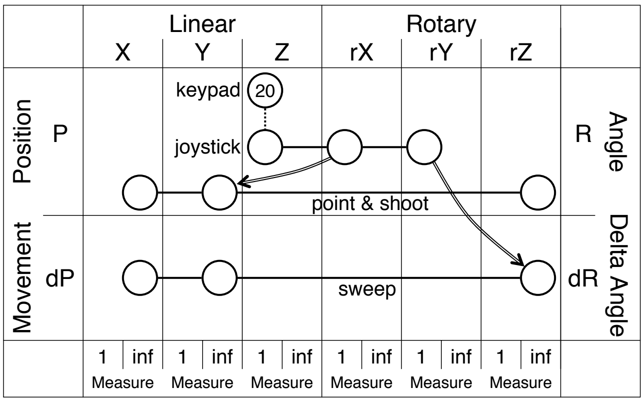


Figure 5: A classification of our mobile phone interaction techniques using Card's design space of input devices [2]. Placing existing phonecam interactions in this design space may help identify promising opportunities for new designs.

around the Y axis (dR:rY) is equivalent to linear X motion. This means that for the *sweep* technique, bending the wrist is equivalent to moving the whole arm. In addition, relative Z movement (dP:Z) could be mapped to a further input dimension. The three horizontally connected circles labeled *point & shoot* represent absolute position sensing. It provides the X and Y position and the state of rotation around the Z axis.

When multiple users interact with a large display simultaneously, multiple cursors are required. This can be achieved by shaping or coloring the cursors differently as done in PebblesDraw [6]. The cursor color could match the shape and color of the cursor on the mobile phone to help users identify which large display cursor they are controlling. Additionally, to help users locate their respective cursor on the large screen, a press on a special phone button could shortly flash or highlight their cursor.

The current implementation of the *point & shoot* interaction clearly has disadvantages for multi-user environments, in that flashing the code grid over the display can disrupt the activity of other users. This problem can be addressed in several ways. First of all, the visual codes can be integrated into the application layout, although this may lower its overall aesthetics. Alternatively, infrared display technology could be used so that they are invisible to the human eye, but still detectable with the camera interface.

3.4 Designing for serendipity

In addition to establishing a coordinate plane, we use Visual Codes (see Figure 2) to encode the public display's Bluetooth address information thus enabling a communications channel to be rapidly established between the mobile phone and the large display. Users merely take a picture of a Visual Code associated with the display and the phone will automatically connect to send (x, y, α, text) information via Bluetooth. The latency to establish the channel is fairly low and the amount of jitter (variance of delay) during interaction is negligible. The same connection can be used to authenticate the user, to send user profiles for adapting the content shown on the large display to personal preferences, to transfer sensitive information to the personal display, and to copy and store information and the current state of interaction on the phone. This creates a very low threshold of use and allows for highly serendipitous interactions. In order to do Visual Code recognition and optical-flow processing, our proposed device interactions require that users install special software on their mobile phone. However, this software could potentially be installed during manufacturing, via the mobile phone network using over-the-air provisioning, or users could retrieve it directly from the computer driving the display via Bluetooth. Fortunately, this software only needs to be installed once and therefore only slightly increases the threshold of use for first time users.

4. IMPROVING PERFORMANCE

We have performed a detailed analysis of these interaction techniques in our previous work [1]. The *sweep* technique performed worse than the *point & shoot* technique and a standard 2D phone joystick for task completion times. One of the limiting factors was that the optical-flow processing has extremely limited capabilities in determining the velocity of arm movement, preventing the use of effective cursor acceleration. This is primarily a problem of perspective, in that the distance from the camera lens to the objects in the image content influences how fast they move in the image picture. Thus, even with perfect optical-flow processing, it is difficult to accurately measure arm velocity. We are currently examining if optical-flow processing can be combined with accelerometers (shown in Figure 7) to produce multi-resolution movement detection where the phone can easily distinguish between fast and slow movements to enable better cursor acceleration. Our hypothesis is that combining sensors will also improve the overall reliability of the movement detection system.

Another problem that we discovered in our user tests was that users made many errors in selecting the appropriate technique. This revealed that our mapping of input techniques to the joystick was hard to remember. The research question emerges of how to map multiple techniques with similar interaction semantics onto the limited interface of a mobile phone.

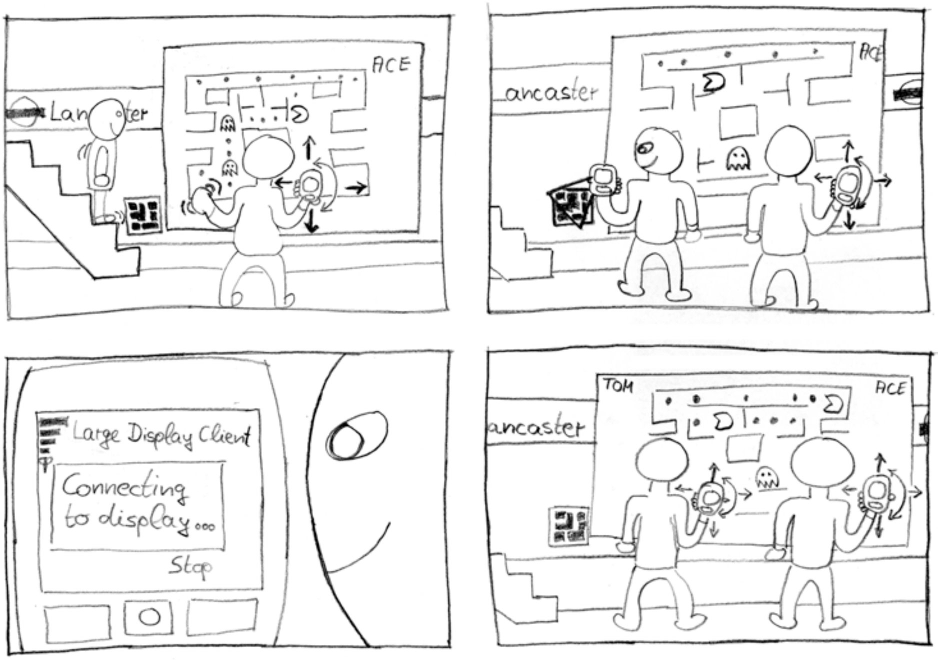


Figure 6: A storyboard illustrating envisioned interactions between mobile phones and large public displays.

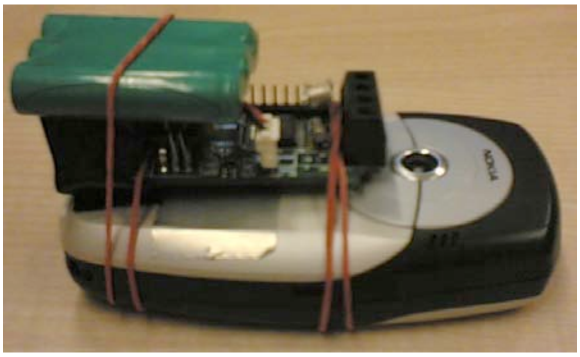


Figure 7: A mobile phone augmented with a 2D accelerometer.

The *Point & Shoot* technique was found to be very sensitive to distance. As users moved away from the display, the targets were perspectively smaller, and thus harder to select. More research is required to determine if zooming lenses or active image stabilization can alleviate these issues. Another approach would be to use hierarchically nested visual codes of several sizes that can easily be recognized from a wide range of distances.

5. CONCLUSIONS

Sweep and *Point & Shoot* enable a new class of highly interactive applications with information terminals or large public displays including interactive art, public games, digital bulletin boards, and advertising. The techniques are functional now, but research is still needed to refine performance and reliability to provide a more pleasant and fluid interaction experience.

6. REFERENCES

[1] Ballagas, R., Rohs, M., Sheridan, J. G., and Borchers, J. Sweep and point & shoot: Phonecam-based interactions for large public displays. *CHI '05: Extended abstracts of the 2005 conference on Human factors and computing systems*. ACM Press (2005).

[2] Card, S. K., Mackinlay, J. D., and Robertson, G. G. A morphological analysis of the design space of input devices. *ACM Trans. Inf. Syst. 9*, 2 (1991), 99–122.

[3] MacKenzie, I. S., and Ware, C. Lag as a determinant of human performance in interactive systems. *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press (1993), 488–493.

[4] Madhavapeddy, A., Scott, D., Sharp, R., and Upton, E. Using camera-phones to enhance human-computer interaction. *Adjunct Proc. of Ubicomp*. Springer-Verlag (2004).

[5] Miyaoku, K., Higashino, S., and Tonomura, Y. C-blink: A hue-difference-based light signal marker for large screen interaction via any mobile terminal. *Proc. UIST*. ACM (2004), 147–156.

[6] Myers, B. A., Stiel, H., and Gargiulo, R. Collaboration using multiple PDAs connected to a PC. *Proc. CSCW*. ACM Press (1998), 285–294.

[7] Rohs, M. Real-world interaction with camera-phones. *2nd International Symposium on Ubiquitous Computing Systems (UCS 2004)*. 39–48.

[8] Silfverberg, M., MacKenzie, I. S., and Kauppinen, T. An isometric joystick as a pointing device for handheld information terminals. *Proceedings of Graphics Interface 2001*. Canadian Information Processing Society (2001), B. Watson and J. W. Buchanan, Eds., 119–126.

Analysis of Built-in Mobile Phone Sensors for Supporting Interactions with the Real World

Karin Leichtenstern, Alexander De Luca, Enrico Rukzio
Media Informatics Group, Ludwig-Maximilians-University Munich
{karin.leichtenstern, alexander.deluca, enrico.rukzio}@ifi.lmu.de

ABSTRACT
There is currently a lot of research going on in the field of mobile interaction with the real world. So far, the environment where the mobile phone is used is mainly perceived as an unpleasant and disturbing factor. Therefore it has rarely been used as a part of the interaction. But on the other hand there is a huge potential for new kinds of the interactions and novel services. Until now, mostly sophisticated and novel hardware has been used for the development of prototypes. In this paper we investigate which sensors are already built-in in modern mobile phones and analyze how they can be employed in real world interactions. Our focus is on investigating how mobile phone sensors can be accessed using the J2ME platform. We analyze the performance and quality of the recorded media data, and where it can be processed. Finally, we conclude with a discussion on what can already be accomplished with today's mobile phones and which new functions are potentially desired.

Keywords
Evaluation, Sensors, Mobile device, Mobile phone, J2ME, MIDP.

1. INTRODUCTION
It is commonly agreed that mobile devices such as mobile phones, smart phones or PDAs have become ubiquitous in our everyday lives. These devices are equipped with a high-resolution color display, they support different standards of wireless networking and they have reasonable processing power and working memory for a great variety of applications. So far the interaction with those devices occurred mostly between the user, his mobile device and the service he uses (e.g. making calls, writing SMS, browsing the web). The usage of the mobile phone takes place in the context of the real world surrounding the user and his devices. This creates challenges, but also offers potential for new services, as well as novel alternatives for interaction. We think that in future the surrounding physical world (people, places, and things [2]) will play an increasingly important role and the interaction takes place between the user, the mobile device, the service and the surrounding world. Recent developments in the field of mobile services have lead to the development of context-aware services that take for instance the location of the user, his preferences or the current time into account.

In addition to this, we see a lot of research projects which involve implicit and explicit mobile interaction with the real world. The presence of the user could be sensed for presenting personalized information on public display or the user could explicitly interact with artifacts in the real world. Most projects in this field were done with enhanced, big and powerful mobile devices such as a Pocket PC PDAs with additional sensors wired to them because the widespread consumer devices, particularly the mobile phones, did not provide enough resources and sensors for their prototypes. But what about the currently available consumer devices which are sold to and owned by the masses? Which sensors are integrated? What is the performance of these sensors? Which of them can be used for the interaction with the real world? How could one program such applications and services that take the existing sensors into account? What programming platform is supported best or by most mobile devices? Which prototypes can be built with these standard consumer devices? This is especially important to the phone manufacturers as no new devices are likely to be sold – the ones that are deployed are awaiting new applications to generate more revenue.

In the Section 2 we show existing projects that used mobile phone sensors. In Section 3 we present a general architecture which is used by most systems that support mobile interactions with the real world and which is also the basis for our tests. This is followed by an analysis how mobile phone sensors can be accessed with J2ME and what levels of quality and performance could currently be achieved. We focus in this paper on J2ME because this platform is supported by most mobile phones nowadays. Based on this, we conclude what is already possible and which requirements are still not fulfilled by the current generation of mobile phones.

2. RELATED WORK
The development and usage of the various sensors in mobile phones was in the focus of a lot of research projects recently. They gathered sensor data for the support of implicit or explicit interactions and for the development of context-aware services and applications [3, 4].

For instance Schmidt et al. [5] used light sensors, microphones, an accelerometer, a skin conductance sensor and a temperature sensor to predict the user's context. They combined the information they got from the different sensor to high-level context information such as "holing the phone in the hand" or "being in meeting". Hinckley et al. [3] used a proximity range sensor, a touch sensitive sensor and a tilt sensor to develop sensing techniques for mobile phones and combined them for recognizing, for instance, if the user picks up the mobile device.

Another application is sensing the surrounding world with the mobile phone. This can be done by sensors such as cameras, infrared sensors, barcodes- or RFID-readers or microphones. Kindberg et al. [2] developed in the context of the Cooltown project the possibility to discover the services that are related to objects in the physical world. They used infrared beacons which can be recognized by a corresponding sensor in the mobile phone. There exist a lot of other projects or products that allow the picking of physical hyperlinks such as Cybercode [6] (camera) from Sony, Airlic [7] (barcode reader) or Near Field Communication [8] (short-range wireless) from Philips.

Most of the previously mentioned work was done with mobile device that were augmented by new sensors, although there are products and implementations which use standard hardware. Rohs and Gfeller [9] used the camera of a standard mobile phone to interpret two dimensional visual codes that represent an ID. Some 505 series models from NTT DoCoMo have a QR Code reader which is based on the camera of these mobile phones.

It has been shown in a lot of prototypes that sensors in mobile phones can lead to new intuitive ways of interactions and to user friendly context aware service. For this, mostly non standard mobile device were used. Some projects already showed that sensors of standard mobile phones can be used for this.

3. GENERAL ARCHITECTURE

In this section we discuss based on the following figure the general relationship between the real world, the mobile phone and the services that are related to the real world objects.

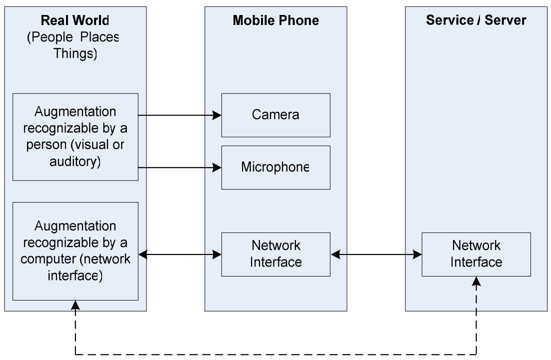


Figure 1. General architecture.

There is no such thing as a definite set of objects in the real world that can be used for mobile interactions. In general, it is possible that any object is augmented (e.g. by a marker or a RFID chip). We focus in this paper on augmented objects because this simplifies the interaction as things like recognizing the name of a building just based on its appearance is still very complicate for specialized hardware. On the one hand, the augmentation could be done in a way that is visible or audible by a human. On the other hand it can be done in a way that is only recognizable by a mobile device. This is a decision that has to be made depending on the actual requirements.

In general, a modern mobile phone has three different built-in sensors: a camera, a microphone and network interfaces (e.g. Bluetooth, GSM, UMTS). For interactions with the real world, one of these sensors must be used to establish a connection between the real world object and the mobile phone. Hereby the real world object could provide static (marker on an advertising poster) or dynamic (public display) information. Most often, interactions are related to a service that is provided by a server or to an application that is already installed on the mobile phone. In principle, it is possible, too, that the real world object augmented by a network interface can directly interact with the service provided by a server.

One possibility for interactions with real world objects is the usage of Bluetooth which is already feasible with the Bluetooth API JSR 82 [13], available on most mobile phones that support J2ME and have a Bluetooth interface. By this it is possible to build interactions with people that are in a defined vicinity to a real world object. New kinds of interactions could be established by exploiting this, but in this paper we focus on the usage of the camera and the microphone.

4. RECORDING MEDIA WITH MMAPI

The platform which is currently supported by most mobile device is the Java 2 Micro Edition (J2ME) [14] with 1.7 billion enabled devices [1]. The main reason is that J2ME is platform independent and nearly all operation systems on mobile devices such as Symbian, Palm OS, Windows Mobile and most mobile phone vendor specific operation systems support J2ME.

The J2ME Mobile Media API (MMAPI) [12] is the only option in J2ME / CLDC / MIDP to address the camera or the microphone of a mobile phone. J2ME is divided in different configurations, profiles and additional APIs. For mobile phones, the configuration ‘Connected Limited Device Configuration’ (CLDC) and the profile ‘Mobile Information Device Profile’ (MIDP) must be used. The basic functionality of the MMAPI is supporting, recording and playing of audio or video data.

In this test we analyzed the performance of the camera while taking pictures and of the microphone while recording audio information. We developed a corresponding J2ME application which can be downloaded at [11]. At the webpage you can also find more test results and samples of the data we got from the sensors. With the programs showed at [11], it is in a first step possible to query which media types in which encodings can be recorded by the mobile phone. Based on this information it is possible to adapt the other programs that test the camera as well as the microphone. These programs were used for the tests explained afterwards. In general, the results of the tests are displayed on the mobile phone screen and/or transmitted to a server for verifying the quality and memory size of the recorded data. We used the mobile phones Siemens S65 and Nokia 6600. We do not compare directly the test results of these two mobile phones because we want to present general results which are valid for more modern mobile phones.

This evaluation discusses the performance of the sensors in a general way, because different mobile phones support different formats and allow different parameter settings. We focus on the formats and parameter settings that are supported by the most devices.

4.1 Image

In this evaluation we concentrated on to test the JPEG format because most mobile phones support this format. Photos stored in the JPEG format are typically used for real life pictures. JPEG pictures offer both a relatively good quality and a moderate memory size. Other formats like BMP or GIF might be more suitable for analyzing the content of the picture but BMP compresses very badly which leads either to a memory consuming or a poor image quality. GIF or PNG should not be used for real live pictures.

4.1.1 *Memory size of pictures*

In the fist test we wanted to determine the memory size and the quality of the taken pictures on a mobile phone in the JPEG format. Particularly for applications in the field of computer vision, like marker detection, a good quality of the taken pictures is necessary. For the first test a J2ME program [11] was implemented to take pictures in different kinds of resolutions. Pictures were taken in resolutions of 80x60, 160x120, 200x150, 320x240, 640x480 and 1280x960 pixels.

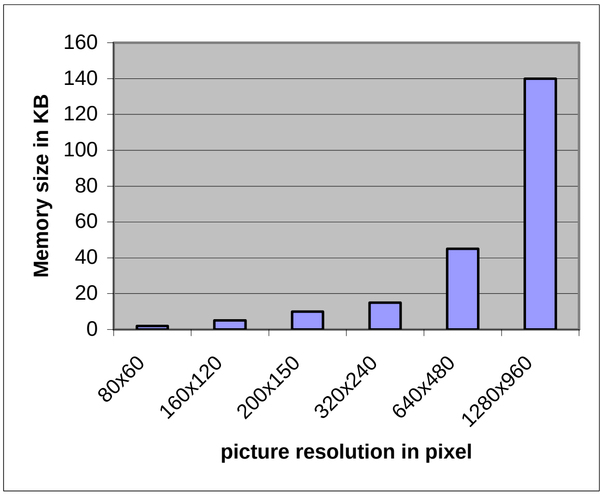


Figure 2. Memory size of different resolution for JPEG.

Figure 2 shows the relatively moderate memory size consumed by the taken pictures. In small resolutions up to 200x150 pixels the memory size did not exceed 10 Kilobytes. Even in the highest resolution of 1280 x 960 pixel the memory size reached only 140 Kilobytes. Beside the moderate memory size the pictures offer also a good quality. Regarding the memory size there was not noteworthy difference between the two phones beside the fact the maximal resolution of the Nokia 6600 is 640x480 pixels.

The quality is already in good region to use the picture for many kinds of detections. The moderate memory size allows to take and to store pictures on the memory card. Depending on resolution, a number of pictures can be stored. Currently J2ME applications for the interpretation of pictures on a mobile phone are relatively slow. Moreover, J2ME does not support some kinds of APIs like Java2D. Therefore an image interpretation, which needs the unsupported APIs, has to be done on a server using e.g. J2SE.

4.1.2 *Speed performance test*

In the second test we wanted to analyze the speed factor. We evaluated how many pictures could be taken in a given time. We decided to take photos in time from one up to ten seconds. In this way we wanted to verify if increasing time changes the number of taken pictures per second. Another variable is the resolution of the taken pictures. We took photos in resolutions 80x60, 160x120 and 200x150 pixel.

The following Figure 3 shows how many pictures could be taken per second in different resolutions. The results shows that as higher the resolution of the picture as decreasing the numbers of taken pictures per second. Moreover, the diagram shows the trend that in the small resolution of 80x60 pixel approximately 1.7 pictures per second could be taken. In the resolutions of 200x150 pixel the trend for the number of taken pictures per second was about 0.8. The overall interpretation of the results is worse. Despite the relatively small resolutions of the pictures the speed performance test shows bad results.

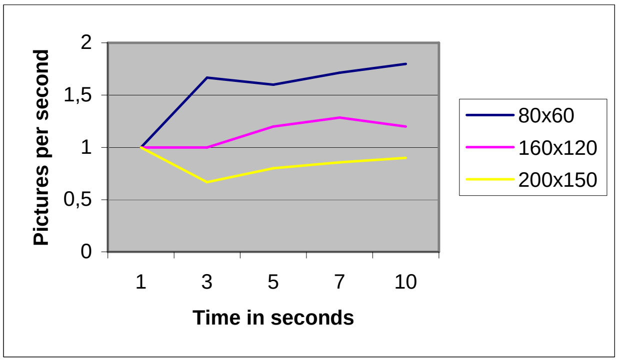


Figure 3. Pictures per second in different resolutions.

Real time applications such as movement detection directly on the mobile phone written in J2ME are currently not possible as not enough pictures may be taken in a given time.

4.2 Audio

In this evaluation we tested the quality of the microphone of the mobile phone when accessed by the MMAPI of J2ME. According to the MMAPI specification audio data in different quality levels from 8 bit / 8 KHz (quality per sample / sample rate) to 24 bit / 96 KHz could be recorded whereby the upper limit makes probably no sense when used with the current microphones because they are constructed for recoding speech. If choosing higher sample rates the file size increases a lot whereas the hearable quality increases only quite a bit.

For the test we used the encoding formats Pulse Code Modulation (PCM) and Adaptive Multi Rate (AMR). PCM is normally used in digital telephone systems. AMR is a lossy Audio data compression scheme optimized for speech coding. These encodings allow multiple applications. AMR for example could be used for voice recognition.

Figure 4 depicts the file size of recorded audio data for 5 seconds when using the encoding PCM and AMR for the bit rates 8 and 16 bit.

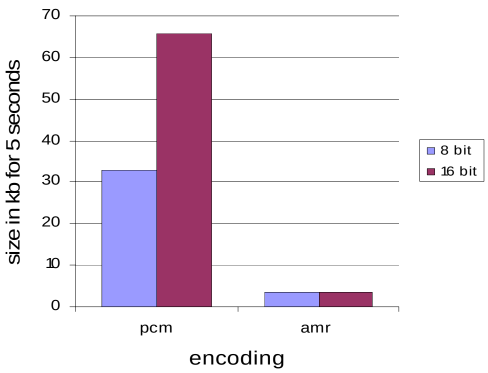


Figure 4. Memory size of different encodings.

The result of this test can be compared to the camera test. The file sizes are in a region that allows in general their processing by the mobile phone. Unfortunately, there is, too, no API available for the interpretation and modification of audio date which postpones this task probably to the server.

The file sizes for the encoding AMR show a typical effect that we recognized particularly during the evolution of the microphone. In principal there are a lot of possible parameters when recording audio data such as encoding, sample rate or bits per sample. Often there exist restrictions regarding these parameters that are not known by the programmer. In the Figure 4 for instance, the mobile phone did support only 8 bit per sample and not 16 bit per sample. This can be only recognized when looking on the file size or by analyzing the generated audio file afterwards.

5. CONCLUSION

In our analysis we found out that in most of the cases the quality of the sensor data is sufficiently good. With respect to the performance, the knowledge about the recorded quality, the processing of the data and the development of platform independent applications for interactions with the real world, there are still considerable problems.

The performance when accessing the camera is currently not sufficient for supporting gestures where about 10 to 15 frames per second are needed because we showed in our test that currently at maximum 2 pictures per second could be recorded.

Another problem is the lack of the knowledge on the quality of the recorded data. When building applications that should run on a lot of mobile devices it is very complicate to test every mobile phone before delivering the mobile phone application or to integrate a test routine that runs after the installation of the program on the mobile phone. It is currently already possible to ask the mobile phone for the supported audio or video encodings. But it would be a great advantage for this kind of programs to know which parameters (e.g. resolution, bit per sample, sample rate) are supported for a specific encoding on a mobile phone. Unless such simple or even more sophisticated methods regarding the quality of service are not present, it is very complicated to develop general programs that use sensor data and which run on several mobile phones.

The next step after recording of the media is the analysis of the audio or video data. Currently there are no APIs available for this task on the mobile phones because J2ME only includes the basic APIs that are needed for the most common applications. Hence the developer can integrate some classes for the interpretation of the sensor data in the program on the mobile device or transmit the data to the server and analyze there the sensor data. The first option has the advantage that no data must be transmitted. But for the processing of the media data on the mobile phone there is often not enough memory or processing power available. The advantage of the transmission of the media data to a server is that a server has a potentially huge set of memory and enough processing power. The disadvantages of this approach are the costs for data transmission and a delay which is not helpful for the most kind of interactions.

Our conclusion is that the development of prototypes for mobile interactions with the real world is already possible when using modern mobile devices whose capabilities are tested intensively before. But we have to wait a few years for matured implementations of the corresponding APIs on the mobile phones, as well as for more processing power and working memory on the mobile phones to support real world interactions.

6. ACKNOWLEDGMENT

The authors wish to express their gratitude to the other participants of the course “programming of applications for mobile devices”, to Nokia and Siemens for lending us mobile phones and to Albrecht Schmidt for their kind support. This work was performed in the context of the framework of IST Project Simplicity funded by the EU [15] and the DFG funded research group “Embedded Interaction” [10].

7. REFERENCES

[1] Report on Heise.de from the 25/10/2004 – News page of a German IT publishing house, http://www.heise.de/newsticker/meldung/52477

[2] Kindberg, T.; Barton, J.; Morgan, J.; Becker, G.; Caswell, D.; Debaty, P.; Gopal, G.; Frid, M.; Krishnan, V.; Morris, H.; Schettino, J.; Serra, B.; Spasojevic, M. 2002. People, Places, Things: Web Presence for the Real World. In: *Mobile Networks* and Applications, 7(5):365–376, 2002.

[3] Hinckley, K.; Pierce, J.; Sinclair, M.; Horvitz, E. 2000. Sensing techniques for mobile interaction. In: *UIST '00: Proceedings of the 13th annual ACM symposium on User interface software and technology*, pp. 91—100, San Diego, California, United States, 2000.

[4] Gellersen, H.; Schmidt, A.; Beigl, M. 2002. Multi-sensor context-awareness in mobile devices and smart artifacts. In: *Journal Mobile Networks and Applications* 7, pp. 341–351, Kluwer Academic Publishersm, 2002.

[5] Schmidt, A.; Aidoo, K.; Takaluoma, A.; Tuomela, U.; Van Laerhoven, K.; Van de Velde, W. 1999. Advanced Interaction in Context. In: HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing, pp. 89-101, Karlsruhe, Germany, 1999.

[6] Rekimoto, J.; Ayatsuka, Y. 2000. CyberCode: Designing Augmented Reality Environments with Visual Tags. In: *Proceedings of DARE*, 2000.

[7] Airlic www.airclic.com

[8] Near Field Communication from Philips Semiconductors http://www.semiconductors.philips.com/markets/identification/products/nfc/

[9] Rohs, M.; Gfeller, B. 2004. Using Camera-Equipped Mobile Phones for Interacting with Real-World Objects. In: *Alois Ferscha, Horst Hoertner, Gabriele Kotsis (Eds.): Advances in Pervasive Computing, Austrian Computer Society (OCG)*, ISBN 3-85403-176-9, pp. 265-271, Vienna, Austria, April 2004.

[10] Research Group "Embedded Interaction", http://www.hcilab.org/

[11] J2ME Application for testing the sensors http://www.hcilab.org/resources/tutorials.htm

[12] Mobile Media API, http://jcp.org/en/jsr/detail?id=135

[13] JSR 82, http://www.jcp.org/en/jsr/detail?id=82

[14] J2ME, http://java.sun.com/j2me/

[15] Simplicity Project, http://www.ist-simplicity.org

Physical Browsing Research

Pasi Välkkynen
VTT Information Technology
P.O.Box 1206
33101 Tampere, Finland
+358 20 722 3353
pasi.valkkynen@vtt.fi

Timo Tuomisto
VTT Information Technology
P.O.Box 1206
33101 Tampere, Finland
+358 20 722 3368
timo.tuomisto@vtt.fi

ABSTRACT
Physical browsing is a mobile-device-based interaction method for pervasive computing. In this paper, we describe our research interests and experiences of physical browsing: the user interaction paradigm, scenarios using physical selection and a demonstration application. We also describe a number of research challenges within the field.

Categories and Subject Descriptors
H5.2. [**Information Systems**]: User Interfaces – *Interaction styles.*

General Terms
Design, Human Factors.

Keywords
Physical browsing, physical selection, ubiquitous computing, user interface.

1. INTRODUCTION
The prevalent visions of ambient intelligence emphasise a natural interaction between the user and the functions and services embedded in the environment or available through mobile devices. In these scenarios, physical and virtual worlds interlace seamlessly, and crossing the border between the worlds appears natural or even invisible to the user. However, a bottleneck to reaching these scenarios can be found in the natural mapping between physical objects and their virtual counterparts. Physical browsing is a means for mapping digital information and physical objects. If physical objects have tags – small and cheap identifiers – attached to them, information in the tags can be read with a suitably equipped mobile phone or PDA, and the functions provided by the objects can be accessed easily and intuitively. This way the physical environment acts as a 'home page' for the services and information provided by the environment and its objects, and the user can access this information by selecting with a mobile device the objects for interaction. Physical browsing is analogous to the World Wide Web: the user can physically select, or "click", links in the nearby environment.

Physical browsing can be divided into steps. First, the user applies *physical selection* for picking the link in the environment with the mobile terminal. The terminal reads the tag and some *action* is launched in the mobile terminal (see Figure 1).

Figure 1. The user selects a link in a movie poster by pointing to it. The mobile terminal reads the tag and the web page of the movie is opened.

Passive RFID tags are a promising and emerging technology for implementing physical selection. These tags are very cheap and get all the power they need from the reader device, which means that it is economically and practically feasible to use them for practically any object we want to access with a mobile terminal. Until recently, the reading range for passive tags has been short. Currently several-meters-range passive tags are being developed concurrently with readers small enough to fit into a mobile phone. Another new technology that is needed for physical selection is interfacing tags with sensors to detect pointing. Ways to integrate optical sensors with passive RFID tags are being researched. The momentum for physical browsing comes from two directions: 1) a need for more intuitive and natural user interaction for mobile applications and 2) the emergence of new technologies – passive sensor-equipped RFID tags and ubiquitous mobile terminals for reading them.

2. RELATED WORK

Want et al. carried out important work regarding the association of physical objects with virtual ones [10]. They built prototypes, some of which were implemented with RFID tags that are read by an RFID reader connected to a PC. Generally their selection method was touching, that is, reading from a short range.

Kindberg et al. built Cooltown [3], an infrastructure over standard WWW technologies, and used it to augment environments like a museum, a shop and conference rooms. They employed short-range RFID tags and infra red communications.

CyberCode by Rekimoto and Ayatsuka [4] is an example of using visual tags. In addition to illustrating the CyberCode as a tagging technology, they described several additional uses for their tags, for example, determining the position and orientation of an object in a 3D space. Additional related work includes GesturePen, a pointing device by Swindells et al. [6], WebStickers by Holmquist, Redström and Ljungstrand [1] and a combination of Bluetooth and passive RFID tags by Siegemund and Flörkemeier [5].

Compared to earlier work, our emphasis in physical browsing is in 1) using passive long-range RFID tags and 2) combining the different selection methods and actions into one unified physical browsing user interface.

3. FROM THEORY TO PROTOTYPE

We have designed and defined user interaction methods for physical browsing and built a prototype that implements those methods. Additionally we have designed several ambient intelligence scenarios that exploit physical browsing as an interaction paradigm.

3.1 User Interaction

One step in physical browsing is *physical selection*, the method by which the user picks with the mobile terminal the tag that he wants the terminal to read – that is, chooses which physical object he wants to access.

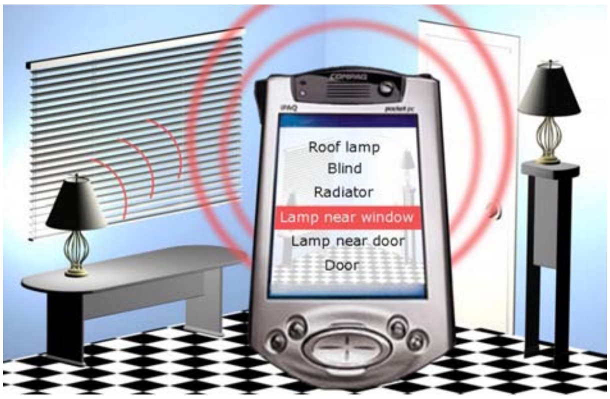


Figure 2. The user scans the whole room and all the links are displayed in the GUI of her mobile terminal.

In *A user interaction paradigm for physical browsing and near-object control based on tags* [9], we defined an information tag and the *PointMe*, *TouchMe* and *ScanMe* methods for selecting information tags by pointing, touching and scanning, respectively. Pointing (see Figure 1) means selecting an object from a distance, by pointing to it with the mobile device. In touching, a tag is selected by bringing the terminal very close to it. Scanning (Figure 2) means reading all the tags in the environment, a useful method if the user does not know where the tags are located or what services are available nearby. Our focus is in RFID tags, but the concepts can be implemented also with infra red and visual tags.

3.2 Scenarios and User Requirements

In a European-Union-funded integrated project MIMOSA[1], we designed several scenarios [2] for ambient intelligence to aid in the development of new sensor-equipped ultra-high frequency (UHF) tags and other MEMS components. These scenarios utilised physical selection as one of their basic user interaction patterns. The vision of MIMOSA is mobile-phone-centric ambient intelligence in which the user uses her personal mobile terminal to access applications and services in the environment. We designed scenarios for everyday applications and four specific application areas: sports, fitness, housing and health care.

3.3 Proof-of-Concept System

We built a proof-of-concept system for UHF RFID tags [7]. The system emulates the predicted sensor-equipped tags that can be read from a several meters' distance and it supports all three selection methods – pointing, touching and scanning. The behaviour of passive RFID tags is emulated with SoapBoxes [8], active wireless components with several built-in sensors. The mobile terminal in our system is an iPAQ PDA equipped with another SoapBox for communicating with the emulated RFID tags. The system was built to demonstrate the feasibility of physical browsing and to act as a tool for studying various usability issues of physical selection and physical browsing.

Figure 3. The components of the mobile terminal for physical browsing. The box on the right is the central SoapBox acting as the tag reader. In the battery case in the middle there is a laser pointer.

Pointing is implemented by using the light sensors of the remote SoapBoxes and beaming them with either infra red or laser light.

[1] www.mimosa-fp6.com

Touching is recognised by proximity sensors. Scanning reads all tags in the vicinity, regardless of whether they are pointed to or whether the reader is near the tags.

To implement pointing with real passive RFID tags they would have to have sensors to detect the pointing beam. This kind of tags – or more specifically a sensor interface for passive UHF RFID tags – is being developed in the MIMOSA project. Since sensor-equipped passive tags are not available yet, we built our emulator to study interaction with them. Even if the emulator system uses SoapBoxes instead of real RFID tags, the user experience is the same as with real tags.

Figure 4. Physical browsing prototype being used for browsing a poster.

We have used posters as demonstration applications (see Figure 4) with several actions embedded into the poster's tags. This specific poster is a conference poster that describes physical browsing traditionally with text and figures. We added four physical hyperlinks into the poster. One of them is a WWW link that contains a URL pointing to our physical browsing WWW pages where the user can get more information. The second link contains a URL that points to an introductory video clip about the topic. Selecting the link opens the video clip in the media player of the iPAQ. The third link is an email address and selecting it opens the email application with the author's address and a predetermined subject already filled in. The fourth link adds a calendar entry for our physical browsing seminar into the iPAQ's calendar application, using a vCard file.

The resources that the tags point to are easily configured so that other applications for different environments can be quickly built. For example, we could build a similar movie poster that contains links to the movie's home page, ticket reservation service (instead of email), a trailer and a calendar reminder for the premiere.

4. RESEARCH CHALLENGES

Physical browsing presents many research challenges, ranging from the user interaction in physical selection to visualising the links in the physical environment. Our main point of view is on the user interaction, but several interesting questions can be raised from other viewpoints.

The contents of the tags have great impact on how and where they can be interpreted. For example, if the tag contains only an ID number, it has to be resolved somewhere into a form that the mobile terminal can use, for example, a URL. If, on the other hand, the tag content is usable "as is", using the tag requires only local connectivity unless it contains a link to the outside world. Optimally, standardisation will provide at least some solutions to these infrastructure questions. The content of the tag also affects the way in which the tag can be displayed in the user's terminal, which directly influences the user experience of physical browsing.

Other important issues that will have impact on the design of physical browsing systems and infrastructures are privacy and security, which will affect the acceptability of applications utilising tags and physical browsing. We will study privacy and ethical issues of physical browsing during our evaluations of MIMOSA scenarios and demonstrations.

In the following subsections we take a closer look at two research challenges we are currently studying.

4.1 User Interface

We are currently continuing work on the user interface for physical browsing. We are evaluating our prototype by conducting user experiments. Some research questions we will look at are the physical properties of pointing and touching (for example, optimal ranges and pointing beam width). Another group of questions is how physical selection should be launched from the mobile terminal. An example of this is whether bringing the mobile terminal close to a tag should be enough for reading it, or should the user also press a button at the same time. Another example issue is the configuration for usable pointing, that is, should the beam be an invisible wide IR beam joined with a laser beam for aiming assistance, or only a narrow laser beam.

We are also studying higher-level user interface issues, such as how to present the results of physical selection to the user. For example, we will look into how results should be displayed if the selection returns zero, one or several read tags, and whether there are differences depending on the selection method and the action provided by the tag. A tag containing a URL could open the URL in the terminal's browser immediately when touched, but what if the user points at a tag and hits another nearby tag as well?

At a later phase we may experiment with different modalities. Speech input, gestures and movements with the terminal could be useful inputs and responses for interacting with the environment via a mobile terminal.

4.2 Visualisation of Physical Hyperlinks

One challenge in physical browsing and especially in the physical selection phase is the visualisation of hyperlinks in the physical environment. Some tagging systems are inherently visual; for example, barcodes and matrix codes must be visible for the reader device to be of any use, and their visual appearance and data content are inseparable. RFID tags, on the other hand, are not necessarily visible because they can even be inserted inside objects. This is sometimes an advantage as the tag will not interfere with the aesthetics of the object it augments, but from the user's point of view it can present a problem.

The main challenges in visualising physical hyperlinks are:

1. How can a user know if there is a link in a physical object or in an environment?
2. How can a user know in what part of the object the link is if he or she knows of its existence?
3. How can a user know what action or functionality will follow from activating the link?
4. How can a user know how a link can be selected if the link does not support all selection methods?

5. PRELIMINARY CONCLUSIONS

In this article we have discussed the concept of physical browsing, a user interaction paradigm for associating physical objects and digital information related to them. As economic and practical tagging systems are emerging, and being integrated with ubiquitous mobile terminals – mobile phones and PDAs – it will be possible to create ambient intelligence settings in greater scale than ever before and truly connect the worlds that have so far been separate, the physical world of our everyday environment and the digital world of the World Wide Web and other information systems.

6. REFERENCES

[1] Holmquist, L. E., Redström, J. and Ljungstrand, P. Token-Based Access to Digital Information. Proceedings of First International Symposium on Handheld and Ubiquitous Computing (HUC) '99, Springer Verlag, 1999.

[2] Kaasinen, E., Rentto, K., Ikonen, V. and Välkkynen, P. 2004. MIMOSA Initial Usage Scenarios. Available at http://www.mimosa-fp6.com.

[3] Kindberg, T., Barton, J., Morgan, J., Becker, G., Caswell, D., Debaty, P., Gopal, G., Frid, M., Krishnan, V., Morris, H., Schettino, J., Serra, B. and Spasojevic, M. People, Places, Things: Web Presence for the Real World. Mobile Networks and Applications, Volume 7, Issue 5 (October 2002). 365-376.

[4] Rekimoto, J. and Ayatsuka, Y. CyberCode: Designing Augmented Reality Environments with Visual Tags. Proceedings of DARE 2000 on Designing Augmented Reality Environments. Elsinore, Denmark. 1-10.

[5] Siegemund, F. and Flörkemeier, C. Interaction in Pervasive Computing Settings Using Bluetooth-Enabled Active Tags and Passive RFID Technology Together with Mobile Phones. Proceedings of the First IEEE International Conference on Pervasive Computing and Communications. 2003.

[6] Swindells, C., Inkpen, K. M., Dill, J. C. and Tory, Me. That one there! Pointing to establish device identity. 2002. Proceedings of the 15th annual ACM symposium on User interface software and technology. 151-160.

[7] Tuomisto, T., Välkkynen, P. and Ylisaukko-oja, A. RFID Tag Reader System Emulator to Support Touching, Pointing and Scanning. Accepted to Pervasive 2005 as a demonstration.

[8] Tuulari, E. and Ylisaukko-oja, A. SoapBox: A Platform for Ubiquitous Computing Research and Applications. Lecture Notes in Computer Science 2414: Pervasive Computing. Zurich, CH, August 26-28, 2002. Mattern, F. Naghshineh, M. (eds.). Springer (2002). 125 - 138.

[9] Välkkynen, P., Korhonen, I., Plomp, J., Tuomisto, T., Cluitmans, L., Ailisto, H. and Seppä, H. A user interaction paradigm for physical browsing and near-object control based on tags. Proceedings of Physical Interaction Workshop on Real-world User Interfaces, September 2003.

[10] Want, R., Fishkin, K. P., Gujar, A. and Harrison, B. L. Bridging Physical and Virtual Worlds with Electronic Tags. Proceedings of CHI 99, Pittsburgh, PA, USA. 370-377.

Task partitioning in smart environments

Markus Ritala
Tampere University of Technology
Institute of Electronics
P.O.Box 692, 33101 Tampere, Finland
+358 50 482 1336
ritala@ele.tut.fi

Jukka Vanhala
Tampere University of Technology
Institute of Electronics
P.O.Box 692, 33101 Tampere, Finland
+358 3 3115 3383
jukka.vanhala@tut.fi

ABSTRACT

This paper studies the use of a Bluetooth enabled mobile phone as a controlling device in a smart environment. Main focus for the work is partitioning of the functionality and the UI handling between a mobile phone and a smart appliance. There seems to be no single right solution for the partitioning of functions. Instead, there are so many case dependent characteristics that the partitioning of functions should be designed in a case-specific fashion. The results of this work give the basis for designing the partitioning.

Keywords

Smart Environment, Distributed UI, Software Architecture, UI Management System

1. INTRODUCTION

The number of smart environments is increasing rapidly [1] and despite the embedded intelligence the users have to be able to control their environment also manually. Although the level of embedded intelligence is increasing, the principal goal should be to activate the user and not to let the home automation take over controlling everything [3][8]. The capabilities of mobile phones have improved significantly in the past few years. Therefore the so-called smart phones offer an interesting opportunity for controlling a smart environment.

It is clear that when there is some smart environment system, there will be appliances from many different manufacturers and those appliances should be able to communicate with each other. They must have the same communication medium, protocols, and semantics of the communication for all the appliances. In addition, when designing the architecture of the control system for the smart environment, an important question is which part of the system has the responsibility for handling which tasks. Here we concentrate on the task partitioning of the User Interface (UI) handling and the functionality of the appliance between the smart environment and the UI device. A Bluetooth (BT) enabled mobile phone is used as a UI device. Also the limitations set by the communication technology are considered. This study has been carried out as a part of the smart environment research at Tampere University of Technology, Institute of Electronics [3]. The results are based on both theoretical considerations and the practical implementations.

At the moment there are two smart home implementations used in this study. The newer one, called eHome [5][6], is a normal two-room flat in an apartment house with paying tenants. The other is a laboratory installation located at the campus and called the Living Room [6][8]. The Living Room includes a hall, a bedroom, a living room, a dining room, a kitchen, a WC, and a sauna. The environment is equipped with several smart appliances from several manufacturers. They are all connected to the controlling system via a cable or a wireless radio link. There are also several different kinds of parallel UI systems [9][10]. New appliances and UI systems can easily be installed despite of the dynamic and distributed nature of the environment.

2. TERMINOLOGY

Before we can discuss about different partitioning solutions, we need to clarify the objects we are talking about. Every *smart appliance* has its characteristic function and it performs *actions*. The abstraction level of these actions may vary a lot. The actions from upper abstraction level may take advantage of actions from lower abstraction level to complete its own tasks.

As an example we may consider a door. Its function is to block the doorway and open to let someone trough. There may be upper abstraction level actions like “open the door” and “close the door”. There may also be actions from lower abstraction level like “turn the door outwards”, “turn the door inwards”, “is the door open”, “is the door closed” and “stop the door”. These lower abstraction level actions could be used to perform previously mentioned upper abstraction level actions. As a result of this, the action that the user thinks as a one simple action may actually be a sequence of many smaller actions. The user usually cares only about actions on the uppermost abstraction level whereas the lower level actions map to hardware capabilities.

In this paper, a collection of those actions that a single smart appliance is capable of performing are referred to as its *functionality*.

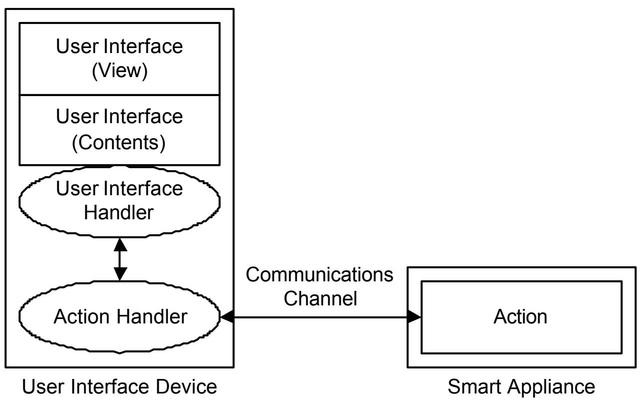


Figure 1: Objects of the control system from upper left corner of the table 2.

Every smart appliance has to offer some way to use its functionality. This is the UI of the device. Here it is referred to as *control interface* to avoid confusion. In an example we used before, the control interface could have two commands. By sending one of those to the door, it will open and using the other the door will close. As we can see, the UI of the appliance does not need to be visible for the human user. It may be for example a list of commands and parameters that are sent to the device trough a communication channel (figure 1).

There may also be description of the view that is meant to be shown to the user. Because the description of the view is not exact, the actual view that the user sees in a controlling device may vary a lot depending on the type of the UI device.

In this paper, the UI is the description of the methods to use the functionality of the certain smart appliance. A *view* is the representation for the user of the UI on some UI device. In the Figure 2 is an example of UI of few appliances. There is also the representation of the UIs, the views on Nokia 7650 mobile phone.

Every UI needs also the logic for handling the UI. The UI handler has to maintain the current state of the UI. Because the screen on a mobile phone is rather limited, in the UI there should be only those functions and parameters that are active for use at the current moment. In this paper the *UI handler* is the part of the system, which can keep the UI updated and display the current UI whenever it is needed.

```
<LIST_ITEM>
2
3
Lower curtain
00 10 60 29 06 BD 01 A0
<END_LIST_ITEM>
<LIST_ITEM>
4
5
Lights on
00 50 37 F9 9D 12 03 A2 0F 04
<END_LIST_ITEM>
<LIST_ITEM>
6
7
Open television
00 02 78 03 F2 C4  01 A2 0F 03
<END_LIST_ITEM>
<LIST_ITEM>
8
9
Close door
00 0E 57 99 A6 D5  06 A0
<END_LIST_ITEM>
```

Figure 2: The view (right) generated from the UI (left) on the Nokia 7650 mobile phone.

3. PARTITIONING OF TASKS

While partitioning the functionality and the UI between the UI device and the smart appliance, there are four situations. It is possible to set both of these tasks for the responsibility of the appliance or the mobile phone. Again, it is possible to divide the responsibility between these parts of the systems such as the other one of those has a responsibility for handling the UI issues and the other one handles the functionality issues.

3.1 UI handling in the smart appliance

When the UI and the UI handling are located in the smart appliance, every time when there is need for the UI information it has to be asked from the device itself. This increases the amount of data that is needed to be delivered in the controlling system. Consequently the need for communication speed is also increased. In addition, there have to be enough resources on the appliance for handling and storing the needed data. Objects of the control system and the interaction between them are shown in figure 3.

For single controlling task there might be several situation where the UI and the description of the view has to be delivered into the controlling device. For example, when we are browsing a menu, the entire user input has to be delivered first into controlling device. The controlling device performs the needed tasks and returns the new UI description. This has to be done in that way because in this solution the controlling device does not know anything about the context of the command. It only acts as a physical UI device and delivers all the user actions ahead. This approach could be called the dumb terminal approach.

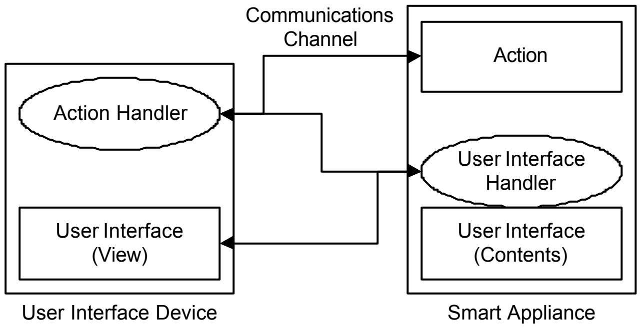


Figure 3: Objects of the control system from upper right corner of the table 2.

A good example of this kind of UI handling system is the prototype of the context aware controlling system made at [11]. In this prototype, the controlling device offers only the physical UI for the user. All the system information is located in the centralized controlling server. The server has responsibility of the UI handling and delivers the current description of it to the controlling device.

3.2 UI handling in a mobile phone

When the responsibility of the UI handling is in the UI device e.g. a mobile phone or a PDA, all the actions that have an influence only on the UI may be done locally in the mobile phone. This solution decreases the amount of transferred data compared with the previous situation. That might be very significant property when the smart environment has several smart appliances and the amount of transferred data is increasing.

Because the computational capacity of the mobile phones is developing rapidly, it is reasonable to centralize the UI processing into these devices. As a result, the complexity and cost of the appliances is lower.

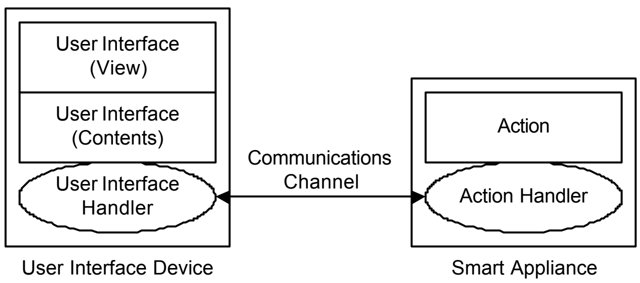


Figure 4: Objects of the system from lower left corner of the table 2.

Centralizing functionality means that the controlling system is much easier to maintain. All the updates are easier to make only to a mobile phone, because the amount of controlling devices are usually lower that the amount of appliances. Also the software development is easier, cheaper and faster for mobile phones than for embedded systems.

It is also possible to create the views into mobile phone that the single smart appliance cannot offer, for example when the function uses several separate appliances to complete its task. This situation from the users point of view is very difficult if appliances handle the UI. Also the location information of the appliance can be easily configured into the mobile phone.

3.3 The functionality in a smart appliance

Setting the functionality into the smart appliance means, that it performs complicated tasks independently using its own basic functions. This partitioning decreases the need for communication between the appliance and the mobile phone. For each controlled task only one or very few control commands need to be sent to the appliance. A single controlling command is very short compared to even a simple UI description.

Again, when increasing the functions of the smart appliance also the complexity of the device increases. This increases the size and of course the cost of the device (figure 5). Secondly it is very hard to combine the functions from separate appliances to perform some more advanced task because the controlling device is not capable to make any decisions and interaction by itself.

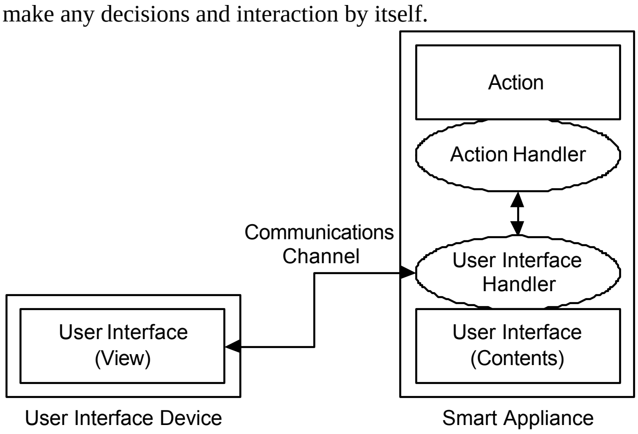


Figure 5: Objects of the control system from lower right corner of the table 2.

However, it is easy to import a new smart appliance into current smart environment because the device includes all the needed functions to work and there are no other requirements for the environment.

3.4 The functionality in a mobile phone

If the mobile phone acting as a controlling device has a responsibility of the functionality of the controlled system it means, that it will perform all the tasks by using the basic functions of the smart appliance (figure 1). As an example we might have a situation where the user wants the car to be heated when he/she is planned to go to work at 7:30 am. In this situation the mobile phone could use an outdoor temperature sensor and a real time clock with the car heater to perform this task. In the Figure 6 there is a simplified graph of the situation. This kind of system uses more communication services than the previously presented system. But because these controlling commands are quite small, the total amount of transferred data is not as large. In spite of that, one controlling situation may need many controlling commands to be delivered to the separate devices and those commands come usually in short bursts, so the communication system should be able to deliver all these command with no noticeable delay.

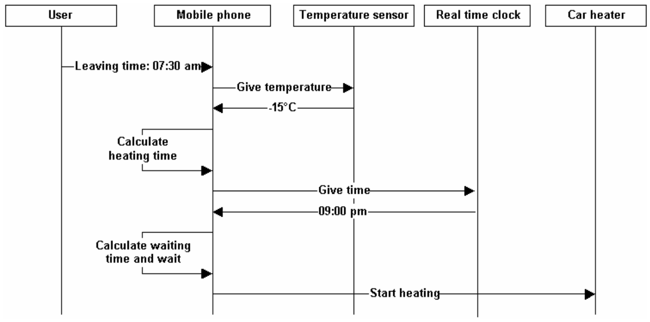


Figure 6: The mobile phone is performing a complicated task by using basic functions of several separate smart appliances.

When we are using the BT as a communication channel, the capacity of the communication will not be the bottleneck with controlling commands. The problems come when there is a need to make a connection between the mobile phone and a smart appliance and after a few commands disconnect and make a connection to another appliance. There can be noticeable delays in connecting and disconnecting the BT devices. In this situation it is important to decide when it is the right time to disconnect.

Partitioning the functionality to the mobile phone, it is possible to take advance of all the mobile phones resources. When the situation is complicated with several devices to control to get the tasks performed, the need of computing and storage capacity increases rapidly.

4. PRACTICAL SOLUTIONS

In this study a simple demonstration of the three out of four possible partitioning situations are implemented (excluding the case in figure 3). Using this demonstration it was easy to find out the characteristics and differences of the situations. The smart appliance used is a simple controllable blinder in the smart environment described earlier. The controlling software was made for the Nokia Series 60 SDK.

Table 1: The controlling commands for controllable blinder.

Function	Description
Blind down	Moves the blind into lower position
Blind up	Moves the blind into upper position
Blind downwards	Moves the blind downwards until it is stopped using separate command
Blind upwards	Moves the blind upwards until it is stopped using a separate command
Stop blind	Stops the blind
Is blind in low position	Tells if the blind is in the lower position
Is blind in up position	Tells if the blind is in the upper position

The blinder can be controlled via a BT connection. A mobile phone acts as a controlling device. The blinder can be set only to upper or lower position. The controlling commands are presented in the table 1. The first two actions are on the upper abstraction level.

These actions are the ones, which the user wants to be completed. These two tasks are actually performed by using the lower level commands.

In the figure 7 there is the construction of the controllable blinder. In the sides of the window, there are guide bars to guide the blind. A magnetic switch tells if the blind is in the lower position. The blind controller and the BT module are behind the window, out of sight.

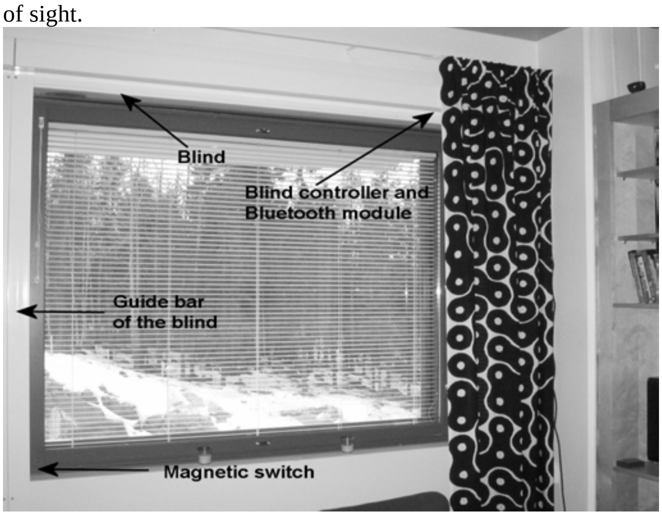


Figure 7: The components of the controllable blinder.

5. CONCLUSIONS

From the further development point of view, placing the responsibility of functionality to the mobile phone brings up many interesting possibilities. One small function of some new appliance may make it possible to implement complex actions. The small sensors like humidity and temperature sensor are the first appliances in a smart environment. When the number of small appliances increases, the new more complex controlling tasks become available by combining these smaller one. Soon only the imagination sets limits for that.

Table 2: characteristics of different partitionings

Functionality \ User Interface	Mobile Phone	Smart Appliance
Mobile Phone	+ Phone characteristics usage + Versatile user interfaces + Combining functions - Sensitive for phone damages - Difficult to add smart appliances	+ Easy to expand + Robustness - Amount of communication - Possible delays - Slowness of connection creation - Hard to implement
Smart Appliance	+ Amount of communication + Easy to control + Easy to implement - Difficult to expand - Difficult to update	+ Clear entirety + Robustness + Easy to add new functions - Complexity of smart appliances - Prize of smart appliances - Limitations of functions

In the Table 2 the characteristics of different partitioning situations are summarized. In the horizontal axis the partitioning situation from the functionality point of view is described and in the vertical axis there are both the UI partitioning situations. This table illustrates how we can in most cases find very different characteristics from the opposite sides of the table.

Partitioning situation in the upper right corner of the table is very difficult to implement in a practice. If UIs are handled in the smart appliance and all the complicated controlling tasks are performed by the mobile phone, it is very hard to create the view for the user. Because a single appliance doesn't know anything about the total controlling situation that its basic function is used for. Therefore in practice, if the functionality is on the mobile phones responsibility, also the UI handling has to be there too.

There is no single right solution for the partitioning of functions. Instead, there are so many case dependent characteristics in every control system and environment that the partitioning of the functions should be designed in a case-specific fashion. The results of this work give a very good basis for designing the partitioning. The study also brings up some issues that could be considered more thoroughly in a further investigation.

The extreme partitioning situations described here above are probably not the best possible for smart environment controlling system. It might be reasonable to use some partitioning between them. That's how we usually get the most suitable characteristics for the system in many cases. For an example we may consider a situation where a UI can be downloaded from the smart appliance using BT. After that the mobile phone has a responsibility of controlling the view and combine the functions of the appliance to get more advanced tasks performed.

6. REFERENCES

[1] Lehto, M., Tekniikkaa ikä kaikki, Finnish Environment 244, Ministry of Environment, Housing and Building, Oy Edita Ab, Helsinki, 6/1998, ISBN 1238-7312

[2] Kaila, L., The Intelligent Home of the Future, Diploma thesis, TUT Electronics, 09.05.2001

[3] Mikkonen, J., Kaila, L., Vanhala, J., "Living Room –project: Towards User Activation", Proceedings of the Interfacing Reality in the New Millennium, OZCHI 2000, Sydney, AU, 12/2000

[4] TUT, Personal Electronics Group, [online, 19.05.3002], URL: http://www.ele.tut.fi/research/ personalelectronics/

[5] TUT Electronics, eKoti, [online, 31.01.2004], URL: http://www.ele.tut.fi/research/personalelectronics/projects/ ekoti_03/index.htm

[6] Koskela, T., Smart Home Usability and Living Experience, Diploma thesis, TUT Software Systems, 04.06.2003

[7] TUT Electronics, Living Room, [online, 26.01.2004], URL: http://www.ele.tut.fi/research/personalelectronics/projects/ smart_home.htm

[8] Kaila, L., Mikkonen, J., Keski-Opas, P., Vanhala, J., "Living Room", Proceedings of the Second International Symposium, HUC 2000, Bristol, UK, 8/2000

[9] Hyvönen, J., Speech control system for intelligent environment, Diploma thesis, TUT Electronics, 28.02.2003

[10] Lehti, L., Älykodin matkapuhelinohjaus, Diploma thesis, Tampere University of Technology, Institute of Electronics, Tampere, 15.01.2003

[11] Ritala, M., Tieranta, T., Vanhala, J., "Context Aware UI System for Smart Home Control", Proceedings of the HOIT 2003, Irvine, USA, 06. – 08.04.2003

Exploring Mobile Phone Interaction with Situated Displays

Keith Cheverst, Alan Dix, Daniel Fitton, Chris Kray, Mark Rouncefield,
George Saslis-Lagoudakis, Jennifer G. Sheridan
Department of Computing, Infolab21
Lancaster University
+44(0)1524 510312

{kc, dixa, df, kray, rouncefieldm, saslisla, sheridaj}@comp.lancs.ac.uk

ABSTRACT
One of the most promising possibilities for supporting user interaction with public displays is the use of personal mobile phones. Furthermore, by utilising Bluetooth users should have the capability to interact with displays without incurring personal financial connectivity costs. However, despite the relative maturity of Bluetooth as a standard and the widespread adoption in today's mobile phones market little exploration seems to have taken place in this area despite its apparent significant potential.

Keywords
Situated displays, Bluetooth, Mobile Phones, Interaction, Prototyping.

1. INTRODUCTION
One exciting avenue for supporting user interaction with situated/public displays is the use of personal mobile phones. 'Historically', one obvious drawback of using mobile phones has been cost, however, by utilising Bluetooth (or its successor protocol) users should have the capability to interact without incurring personal financial connectivity costs. However, despite the relative maturity of Bluetooth as a standard (it was actually standardized in 1996) and its widespread adoption in today's mobile phones little exploration seems to have taken place in this area despite its apparent significant potential.

In this paper we introduce our current work on exploring this area. The work described is being carried out under the auspices of the EPSRC funded CASIDE project (Investigating Cooperative Applications in Situated Display Environments, see: www.caside.lancs.ac.uk/ for further details).

2. CURRENT EXPERIMENTS – SITUATED DISPLAY DEPLOYMENTS
In the last three years we have deployed a small but significant number of situated displays within the Computing Department of Lancaster University. When we refer to such situated displays we agree strongly with the definition provided by O'Hara [8]:

> *In recent years, more and more information is being presented on dedicated digital displays situated at particular locations within our environment. At their most basic, digital display technologies allow information to be more easily updated dynamically and remotely. However, these new kinds of interaction technologies also allow people to use these situated displays in novel ways both as for the individual's purposes and in the support of group work.* (O'Hara et al. 2002).

The following two subsections describe two of these existing deployments and our initial experiments to investigate how Bluetooth technology might provide new and useful ways for users to interact with the deployed displays using their own personal mobile phones.

2.1 Interaction with Office Door Displays
The first Hermes door display (see figure 1) was installed outside one of the offices in Lancaster's Computing department in March 2002 and by October 2003 the number of deployed units had increased to ten. Unfortunately a move of building meant that the Hermes system (see [4] and Chapter 6 of [9] for more details) had to be dismantled but during the 24 months of its use 775 notes were left by visitors and over 5278 messages were set by owners.

Figure 1: An early Hermes display.

The system is currently being redeployed in the new office building and we are currently experimenting with new design ideas. For example, in the initial Hermes system each office only had one display - we are trialling the use of two displays per office. This would allow one display specifically for supporting interaction with the visitor to an office, e.g. enabling the visitor to leave a note for the owner. The other display would be solely for the owner leaving messages to visitors.

One concern with the original system was security – we did have 2 units stolen outside of office hours. One potential benefit of Bluetooth is that users can interact with the system, e.g. to leave a note for the owner of the door display, while the display is more safely secured within the office.

Currently it is envisaged that visitors will be able to use the system in order to download, via Bluetooth, information from the office door display such as the owner's contact details. However, studies have shown that owners are prepared to share certain information when a student has made the journey to their office which they would not necessarily make available on, for example, their web home page. We plan to investigate this issue more fully and ascertain whether the owner of a door display is prepared to share and make available objects such as calendar details based on the scoping effect resulting from the limited range of the Bluetooth signal – in effect restricting the availability of such information to persons that are within close proximity to her actual office (see figure 2a). Figure 2b shows the kind of message that could be 'scribbled' and then sent by an owner or visitor using a mobile phone equipped with a touch screen.

Figure 2a (left): Visitor using his mobile phone to interact with a Hermes display, and, figure 2b (right) the kind of display that could be composed using the jotter app found on touch-screen based phones such as the Sony-Ericsson p800.

2.2 Interaction with Public Photo Display

We first deployed the Hermes picture display in June 2003. The system is effectively an extension to the Hermes system and was first designed to enable users on a particular corridor to send pictures through MMS, e-mail etc.(see figure 3a) - in the subject header of the message the user would stipulate the location of the display, e.g. "PUB LOC C FLOOR". The display is actually a Phillips 'smart' display and in addition to displaying a picture presentation the system was recently extended to support a presentation to show the vCards of users that have 'registered' with the display (see figure 3b).

Figure 3a (left): the Hermes Picture Display showing the picture presentation, and 3b (right) showing the registered users' contact details presentation.

Clearly with MMS we already support some level of interaction with the photo display – but we are currently experimenting with ways in which users can interact when co-located with the display via a Bluetooth connection to their mobile phone. To this end we are investigating a range of interaction and user interface ideas.

2.2.1 Asynchronous Interaction

A user can use the built-in application on his or her mobile phone in order to send or receive a picture or contact details in vCard format (RFC 2425) from the photo display over Bluetooth.

Figure 4 below illustrates the photo display and a user sending a file (in this case it happens to be a picture) via their phone to the display. As a prerequisite to sending a file the user must firstly pair with the display - this involves searching for the display (see figure 5a) and then entering the appropriate pass key (note that details of this appear in the centre of the presentation).

Figure 4: Visitor using his mobile phone to interact with the Hermes Picture Display using Bluetooth.

In order to select a picture or vCard for download onto a mobile phone a variety of approached are possible. For example, a registered user may simply touch the display to select a picture or vCard and then select their (discoverable) Bluetooth phone from a list in order to initiate the transfer.

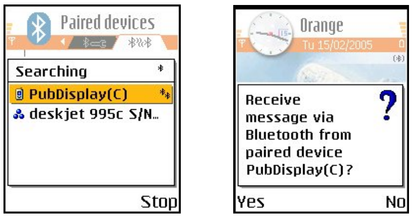


Figure 5a (left): Visitor waiting to pair with the public display, and, figure 5b (right) user accepting to receive selected vCard file.

However, requiring a user to touch the screen necessarily restricts the number of users that can select a picture concurrently (however, in practice this might provide an interesting opportunity for social engagement!). So with the aim to investigate how we might support a group of users interacting with the display (for example during a departmental research open-day) we are also considering more synchronous interaction approaches. Note that at this stage we are considering how to support simultaneous interaction with a small group (i.e. of no more than 8 people) – clearly other approaches would be required to support scalability up to an arbitrary number of users but it is certainly not clear that such scalability is required at this stage nor that such highly scalable solutions would be appropriate for the smaller sized group numbers envisaged for this particular deployment.

2.2.2 Synchronous Interaction

A number of approaches are possible, some more engaging and visible than others and it is our intention in the coming weeks to carry out (initially) simple user studies in order to find which methods users prefer. The remainder of this section considers the selection and download of vCards from the display.

One implication of supporting a more synchronous interaction style is that the user must first download and install a Java application (see figure 6 below) and one of our early studies will be to determine how users feel about this, e.g. in terms of issues such as trust etc.

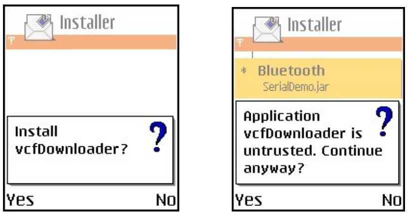


Figure 6: Downloading a Java application to support a more synchronous interaction style.

2.2.2.1 Method A – Use of colour on the display

On pairing with the display the user downloads a small Java application. This application assigns the user a colour and this colour is used to highlight the image currently selected by her. The user can use keys/joystick on her mobile phone to select the appropriate vCard and she can press the OK button in order to receive the picture. Sample screen shots showing how the display would appear with two different users is shown below in figure 7.

Figure 7: Two different approaches for annotating the presentation to show images selected by different users.

2.2.2.2 Method B – Simple use of Id's

This approach is simpler but may not necessarily be a more pleasing method for the user. Again the user would download a small application and would then press the appropriate button to select and retrieve the required vCard.

Figure 8: Annotating the presentation with digits to enable a user to select a vCard for transfer.

2.2.2.3 Method C – GUI on the mobile phone

This approach would take advantage of the mobile phone's colour graphical display. The user again downloads a java application which in this case displays a set of thumbnails on the phone – the user can then choose the image relating to the vCard that she wishes to have added to her contacts list.

Figure 9: Exploratory UI design using J2ME wireless toolkit (note that the positions of icons correspond to those that appear on the presentation appearing in figure 3b).

3. Current State of Development

Having completed development of the asynchronous approach, we are currently developing the java applications to enable the trial of the synchronous approaches. This is being achieved using Linux servers running an Open Source Bluetooth stack These applications will utilize the RFCOMM transport - one of the higher levels of the Bluetooth stack - which emulates asynchronous serial port connections over a Bluetooth link. RFCOMM sits on top of the L2CAP (Logical Link Control and Adaptation) layer (providing connection and multiplexing of higher level protocols) which in turn sits on top of the lower hardware oriented layers.

4. Supporting Community - On and Off Campus

The overriding aim of the CASIDE project is to investigate how the deployment of situated displays can support the notion of community, in both campus and other settings. However, situated displays do not typically fit the traditional single user mouse/keyboard interaction style. We will seek to explore the interactions that manifest themselves (over time) in a range of settings both on and off campus.

Much of this exploration will be guided based on our understanding of the settings and will utilise techniques found in context-aware computing (location-aware behaviour, automatic personalisation/content creation based on sensed context, etc.) and tangible interfaces as well as more familiar modalities such as e-mail, instant messaging and mobile phones.

Our approach will be based on a combination of theoretical research, collection of empirical data sets (e.g. arising from use of cultural probes [6]) and prototyped application development.

This methodology involves a tight cycle where theoretical issues and understanding, developed through reflection on empirical observations, are used to design deployed systems that test and explore the theory. These deployed systems then create a new context for observation of user behaviour and thus lead to fresh insights, discoveries and refinement of theoretical understanding.

As is evident, and has been noted previously, a central aspect of this methodology is the deployment of systems as technology probes. In order to achieve real use, these systems must do more than just explore interesting issues they must also meet real or emerging needs. We will therefore adopt an iterative and participatory design approach to each of our deployments where the observation and involvement of users will serve the dual purpose of traditional user-centered design and source for more theoretical analysis.

4.1 Deployments within Campus

On campus we plan to explore how the styles described above can support interaction, e.g. the football or climbing society could have displays situated alongside their existing more traditional notice boards (see figure 10 below). Walking past a display could serve to prompt the player of a football team to send pictures or video footage from their mobile phone of a game they watched over the weekend. He or she may then use their phone to download a match report that had been posted previously.

Figure 10: Notice board outside the University Climbing Wall.

4.2 Deployments Off Campus – Domestic and Residential Care Settings

Outside of the campus setting we intend to investigate how public displays can be used in care settings. This follows on from previous work (see [3]) but here we hope to explore how these technologies can support a sense of community. For example, a recent design workshop revealed the potential for using a display situated in the common room of a residential care facility in order to support a sense of community between both residents and staff. When deploying technologies in such settings it is crucial that the deployed systems are reliable – effectively the early studies/deployments described above can serve as an excellent means for 'burn-in' testing of the technology solutions.

5. RELATED WORK

There is surprisingly little published work relating to the combination of mobile phones, situated/public displays and Bluetooth. One exception is the work on ContentCascade [7] which enables a user to download content from a public display onto her mobile phone using Bluetooth. The system was tested in a small and informal user study using movie clips. The ContentCascade framework enables users to download either summary information or the movie clips themselves.

However, there is now a reasonable body of research on situated display technologies – and a good survey of this can be found in [9]. The WebWall is a system which enables multi-user communication and interaction via shared public displays, e.g. airports, [5]. WebWall allows pervasive and seamless access to the web-based application such as simple sticky notes, and image galleries via devices such as mobile phones or PDAs. WebWall's architecture enables a strict separation of I/O technologies (like HTTP, email, SMS, WAP, MMS etc.) from components managing storage, presentation logic and physical display technologies. While WebWall would clearly be able to incorporate Bluetooth, it is less clear how the special features of Bluetooth could be used in the current architecture.

Other examplesof work describing deployed systems that utilize the combination of mobile phone and situated displays with SMS include the txtBoard system [1] developed by the Appliance Studio, for example, have developed an Information Appliance - this supports 'texting to place' and has the family home as its primary deployment domain. In [11] the authors describe the short term trial of a system supporting the sharing of pictures which utilises a laptop-sized display situated in the family's living room. In terms of work describing phone/display interaction based on visual codes, one interesting approach is described in [2]. An interesting potential approach for the pairing of devices, e.g. mobile phone and situated display, could be 'SyncTap' [10].

6. SUMMARY

In this paper we have introduced our current work on exploring the use of Bluetooth equipped mobile phones to support interaction with situated displays. Our approach is to produce prototype deployments and involve potential users at an early stage. In the near future we hope to take our designs beyond the university domain and into a residential care facility.

7. ACKNOWLEDGMENTS

This work is funded by the EPSRC funded CASIDE project (grant ref: EP/C005589). The work also builds on work carried out under the EPSRC funded Equator and CASCO projects. Hardware support for future trials based on the work described will be provided by the SRIF II (Science Research Investment Fund) funded e-campus initiative.

8. REFERENCES

[1] Appliance Studio (2002). *Press Release for TXBOARD*: http://www.appliancestudio.com/about/press/20021203.pdf

[2] Ballagas, R., Rohs, M., Sheridan, J.G., and Borchers, J. Sweep and point & shoot: Phonecam-based interactions for large public displays. To appear: CHI '05 Extended abstracts, ACM Press (2005).

[3] Cheverst, K., Clarke, K., Fitton, D., Rouncefield, M., Crabtree, A. and Hemmings, T. "SPAM on the menu: the practical use of remote messaging in community care", *Proceedings of the 2003 ACM Conference on Universal Usability*, pp. 23-29, Vancouver, Canada: ACM Press. 2003

[4] Cheverst, K., Dix, A., Fitton, D., Friday, A. and Rouncefield, M. "Exploring the Utility of Remote Messaging and Situated Office Door Displays", *in Proc. of the fifth ACM International Symposium on Human Computer Interaction with Mobile Devices and Services*, Udine, Italy, LNCS, Springer-Verlag, pp. 336-341, Sept. 2003.

[5] Ferscha, A. and S. Vogl, Pervasive Web Access via Public Communication Walls: Pervasive Computing, Springer LNCS 2414, Zurich, Switzerland, 84-97, 2002.

[6] Gaver, W., Dunne, A., Pacenti, E. Design: Cultural Probes, in *Interactions: New Visions of Human-Computer Interaction*. 6(1), pp. 21-29. (1999).

[7] Himanshu Raj, R. Gossweiler, D. Milojicic, "ContentCascade Incremental Content Exchange between Public Displays and Personal Devices", in Proc. of the firsy Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'04), Boston, Massachussets, USA, 374-381, 2004.

[8] O'Hara, K. E. Churchill, M. Perry, D. Russell, N. A. Streitz, *Public, Community and Situated Displays: Design, Use and Interaction around Shared Information Displays*, www.appliancestudio.com/cscw/cscwdisplayworkshopcall.htm (2002).

[9] O'Hara, K. M. Perry, *et al Public and Situated Displays: Social and Interactional aspects of shared display technologies*, Kluwer. ISBN 1-4020-1677-8. 2003.

[10] Rekimoto, J. Ayatsuka, Y., Kohno, M. "SyncTap: An Interaction Technique for Mobile Networking", *in Proc. of the fifth ACM International Symposium on Human Computer Interaction with Mobile Devices and Services*, Udine, Italy, LNCS, Springer-Verlag, pp 104-115, 2003.

[11] Romero, N., Baren, J. v., Markopoulos, P., Ruyter, B. d. and IJsselsteijn, W. Addressing Interpersonal Communication Needs through Ubiquitous Connectivity: Home and Away, in *Proc. of European Symposium on Ambient Intelligence*, pp. 419-429. 2003.

Motion Detection as Interaction Technique for Games & Applications on Mobile Devices

Stephan A. Drab
Upper Austria University of Applied Sciences
Mobile Computing
Hauptstrasse 117
Hagenberg, Austria
stephan.drab@fh-hagenberg.at

Nicole M. Artner
Upper Austria University of Applied Sciences
Media Technology and Design
Hauptstrasse 117
Hagenberg, Austria
nicole.artner@fh-hagenberg.at

Keywords
mobile devices, pervasive, interaction techniques, motion detection

ABSTRACT
Mobile devices become smaller and more powerful with each generation distributed. Because of the tiny enclosures the interaction with such devices offers limited input capabilities. In contrast there are hardly any mobile phones purchasable that do not have a built-in camera. We developed a concept of an intuitive interaction technique using optical inertial tracking on mobile phones. The key of this concept is the user moving the mobile device which results in a moving video stream of the camera. The direction of the movement can be calculated with a suitable algorithm. This paper outlines the algorithm *Projection Shift Analysis* developed to run on mobile phones.

1. MOTIVATION
Our approach detects the relative motion of the camera in the two-dimensional pixel space of the image. As camera movement directly results in motion of all scene components the motion of the camera movement can be defined as the inverse movement of the scene. If there are no significant scene components moving for itself conventional motion detection methods can be used to analyse the video stream. There are several algorithms from different fields of computer graphics and image processing used to parameterise the motion of the scene. Although they pursue different approaches, all of them would analyse the scene motion sufficiently. Due to low CPU and memory resources on mobile phones we developed the *Projection Shift Analysis* algorithm for motion analysis. There is a wide range of applications for the motion parameter, for example controlling a game similar to joystick interaction. Another possible application could interpret motion gestures or control the cursor like the stylus input technique on PDAs.

2. RELATED WORK
In [8] Geiger et al. present concepts for mobile games and address interaction issues concerning mobile devices.

The simplest approach to track objects is detecting significant features in an image, e.g. edges. Naturally, edge detection methods like the Robert, Prewitt, Sobel, Laplace or Canny [2] filters are used to achieve this.

Motion Detection in 3D-Computer Graphics, Mixed- and Augmented Reality is often referred to as *Tracking*. Beier et al. presented a markerless tracking system using edge detection in [1]. Comport et al. propose a robust markerless tracking system in [3]. A specialised solution to the problem of markerless tracking was published by Simon et al. in [15]. In [11] Moehring et al. present a marker-based tracking system designed especially for mobile phones. Kato and Billinghurst developed the optical marker based tracking system ARToolKit published in [9].

Foxlin et al. present a wide spectrum of optical inertial tracking systems in [5, 6, 17]. Additionally, a taxonomy of Motion Detection methods has been published in [4]. In [10] Koller presents a method to track the position of cars using an optical system.

Siemens Mobile developed a game called *Mozzies*, that is distributed with the mobile phone SX1 by default. This Symbian based game augments the background video from the camera with moths. The user can point the gun at a moth and shoot it by moving the phone and pressing the appropriate button. In [7] Geiger et al. present an interesting approach of an augmented reality version of a soccer game running on a PDA.

3. INTERACTION TECHNIQUES
This chapter describes and classifies various interaction techniques used on mobile devices nowadays. There are a few main parameters that define the usability of those techniques. The **reaction time** between the user input and the response on output devices such as the display is a very crucial parameter. Any visual response on the output device after about 200 ms is not interpreted as a direct reaction to the user, but as a separate event. The **quantity** of actions a user is able to perform using a specific input technique defines the speed at which he can interact with the device. The **intuitivity** of an input method strongly affects the usability

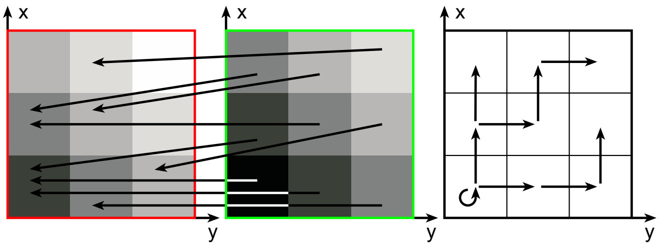


Figure 1: An example of the Block Matching algorithm.

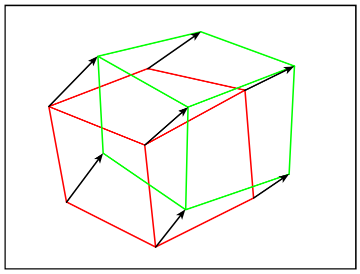


Figure 2: Edge Detection and Tracking sample.

and user acceptance.

The following interaction techniques for mobile phones were evaluated regarding the above parameters. **Keys** are the most common input technique for mobile phones. They offer a very short reaction time but lack intuitivity. The rate of interaction is dependent on the user's experience with this technique. **Voice recognition** is a very intuitive input technique, but lacks fast reaction time and input quantity. **Touch screens** are the most intuitive way of interacting with a screen based mobile device. The reaction time is comparable with key input but the input quantity strongly depends on the GUI design. Unfortunately aside from smartphones there are no mobile phones featuring a touch screen.

Each of these techniques has mentionable advantages. Key interaction offers a good reaction time. Voice recognition is very intuitive to use. Touch Screens combine a fair amount of intuitivity and interaction quantity, but aside from smartphones they are not available on mobile phones. Thus we developed a concept and implementation for an interaction technique particularly for mobile phones that incorporates the advantages of all presented techniques.

4. MOTION DETECTION ALGORITHMS

The term *Motion Detection* describes a set of algorithms that detect the motion in a successive image sequence, e.g. in a video captured by a camera. In this section we present suitable motion detection algorithms:

4.1 Block matching

Block matching is a method from the video compression sector. It divides the images in equally sized blocks to find the best matching block in a reference picture for all blocks of the current image. It yields to describe an image not using color values per pixel, but as block references to the previous image along with other parameters, as Richardson describes in [13].

Even tough the algorithm has not been developed to analyse motion in an image sequence, it can be extended to fit this purpose. The algorithm determines where a block of an image will be positioned in the next image. The resulting block references can be interpreted as motion vectors in units of blocks. An arithmetic middle of all block motion vectors multiplied by the pixel size of a block would lead to a single motion vector describing the approximate relative motion.

In figure 1 a sample of a matching procedure is depicted. The frame on the left shows the reference image. In the middle the current image is depicted which is encoded using block references to the previous image. The arrows in between indicate the block mapping information calculated by the algorithm. In the right frame the block references are vizualised as motion vectors. The arithmetic middle of the motion information from the right frame in figure 1 would result in a vector of $(\frac{8}{9}, \frac{8}{9})$ which almost matches the relative movement of $(1, 1)$ in our example.

4.2 Edge Detection and Tracking

Edge Detection and Tracking is used throughout markerless *3D-Tracking* algorithms[1]. In the first step the edges are extracted from the image. This is possible through either folding the image with an appropriate *Edge Detection* matrix [2, 14] or using the *Hough Transformation* [16]. The detected edges can then be tracked throughout the image sequence. Figure 2 shows an example of detected edges of a cube that have been tracked in two images.

The edge comparison step in the Edge Detection and Tracking algorithm is applied to an unpredictable high number of edges. Thus the processing time used for this operation cannot be forecast and can become a bottleneck in case the images depict a huge number of edges. Although Edge Detection and Tracking is the most promising algorithm, it is not capable of running on devices with low computing power at a reasonable frame rate.

4.3 Analysis of Scene components

Moving scene components in a video stream result in a partial motion of a scene. In [10] Koller describes an algorithm to extract motion information of scene components from an image sequence. The first step in this approach extracts the objects' parameters from the image. Thereafter the proximate image is searched for objects and relative motion in pixel space is calculated. An example is depicted in figure 3. Again, an arithmetic middle of the motion vectors of all scene components could be used to determine the scene and camera motion.

The analysis of the motion of scene components is similar to *Edge Detection and Tracking* and incorporates the same disadvantages like unpredictable computing time and consuming high computing power. Therefore, it cannot be used on mobile devices.

[1]See [3, 15] for examples of 3D Tracking algorithms.

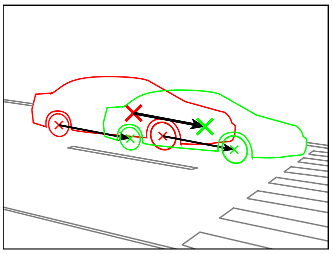


Figure 3: Scene Component Analysis sample.

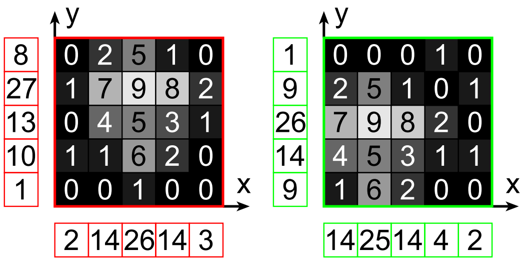


Figure 4: Horizontal and Vertical projection buffers for each of the two successive images.

5. PROJECTION SHIFT ANALYSIS

The previously presented motion detection algorithms qualify for detecting motion information in an image sequence using a high amount of calculating time. However, in mobile applications computing power is a rare resource, therefore we propose a new method called *Projection Shift Analysis*. The algorithm does not require color information, but grayscale images as input. Fortunately, most cameras support captured images in the YUV format[2] which holds a separate luminance (Y) channel containing grayscale information. This algorithm will discard the provided color information (U and V).

5.1 Image Projection

In the first step the image is projected onto its x- and y-axis. That means all luminance values of each row is summed up in the vertical and each column in the horizontal projection buffer as depicted in figure 4. The horizontal and vertical projection buffers $pbh(x)$ and $pbv(y)$ of the image $im(x, y)$ with the corresponding width w and height h are defined as $pbh(x) = \sum_{i=0}^{h-1} im(x, i)$ and $pbv(y) = \sum_{i=0}^{w-1} im(i, y)$ with $x \in [0..w[$ and $y \in [0..h[$.

5.2 Shift Analysis

If the scene within an image sequence is moved vertically or horizontally the corresponding projection buffers will shift equally. For example an image sequence containing a horizontal panning shot from the left to the right will result in a horizontal projection buffer whose values will shift to the left over time. Thus our approach searches for the best matching shift between two projection buffers of two successive images. Figure 5 shows three examples of different shift values.

To estimate the best match we introduce a value called *badness factor* that is similar to a correlation factor and characterises the badness of a shift. A lower value indicates a higher probability for this shift to be correct. The algorithm calculates the badness factor for every possible shift through summing up all squared differences of the values of the compared projection buffers. Because the calculated badness factors result from a different number of compared values they are normalised by dividing it by the number of values compared. Figure 5 shows the calculation of badness factors using the vertical projection buffers of the image displayed in figure 4.

[2]For more information on the YUV format we refer to [12].

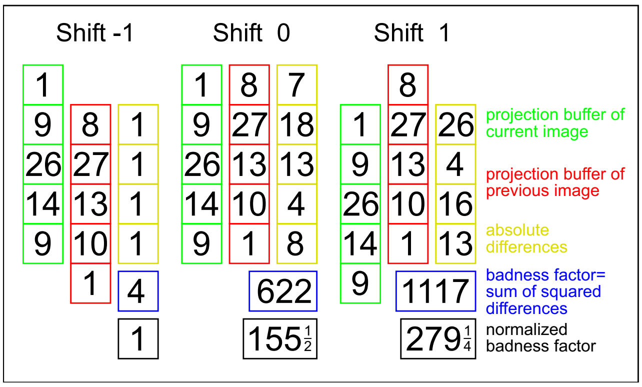


Figure 5: Example of badness factor calculation for different shifts of two projection buffers.

The shift value used to calculate the smallest normalised badness factor is assumed to be the correct relative motion between the two compared images. In our example depicted in figure 4 the best matching shift value of -1 exactly matches the vertical movement of 1 pixel. This procedure can be applied to horizontal and vertical projection buffers equally to estimate the motion in x- and y-axis.

6. RESULTS

Tests determining the robustness of the algorithm *Projection Shift Analysis* elicited the following restrictions:

If there are *significant scene components moving* in the captured video stream the motion detection often "follows" these objects instead of detecting the motion of the background. *Repeating patterns* – for example a chess board – often lead to jumping motion detection results. The color values of the analysed images need to have a *wide dynamic range*. Most cameras of mobile phones use a relatively *long shutter time* due to low quality lenses and CCD chips. If the camera moves too fast this can result in blurred images which cannot be analysed correctly. The quality of the detected motion parameters decreases if a *lower frame rate* is used. The camera used to capture the video stream must support a *sufficient resolution and image quality*. The algorithm does not support detection of *rotations* around the camera's view axis. However, unusable motion detection results are only produced in some cases.

To further examine the robustness two application prototypes were developed to evaluate the functionality and usability of the proposed concept:

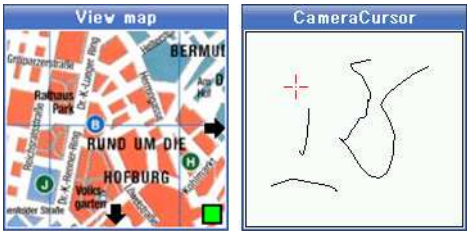


Figure 6: The prototypes MAPnavigator and CameraCursor.

MAPnavigator is an application that allows the user to view different maps. The problem in this context is the low display resolution of mobile phones. Consequently it is not recommended to scale down a huge map because then the user is not able to make out important details in the map. The solution to this problem is to divide the map into several parts. MAPnavigator uses the *Projection Shift Analysis* motion detection algorithm to seamlessly navigate through the map. It is very easy and intuitive to find the requested part of the map using the motion detection technique. The left part of figure 6 shows a screenshot of the application.

The application **CameraCursor** was developed to test if our motion detection algorithm is capable of controlling a cursor. The user is able to control the cross cursor by moving around the mobile phone. By pressing and holding the appropriate button the cursor draws lines while moving on the screen. Additionally the speed of the cursor can be configured at run time. An example of the application is shown in the right part of figure 6.

7. CONCLUSIONS

The *Projection Shift Analysis* algorithm works if none of the given cases in section 6 occur. Compared to the algorithms in section 4 it uses very little CPU and memory resources. Therefore it is ideal to be used on mobile phones. In case the camera is moved quickly the results may be inaccurate by a few pixels. Using the motion parameters as user interaction does not require high exactness for fast movements because the user does not have an indication of inaccuracy as long as the movement displayed on the screen is approximately following his motion. However, there are some cases the algorithm cannot keep track of the motion and fails completely. This mostly happens when the user quickly moves the camera while pointing at dark spots or repeating patterns. Future implementations are planned to filter out motion noise using the *Kalman Filter* which would also reduce jumping motion parameters caused by unfavorable environments.

8. REFERENCES

[1] D. Beier, R. Billert, B. Brderlin, D. Stichling, and B. Kleinjohann. Marker-less vision based tracking for mobile augmented reality. In *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 258–259. IEEE, Inc., 2003.

[2] J. F. Canny. Finding edges and lines in images. Master's thesis, MIT, 1983.

[3] Andrew I. Comport, Eric Marchand, and Francois Chaumette. A real-time tracker for markerless augmented reality. In *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 36–45. IEEE, Inc., 2003.

[4] Eric Foxlin. *Handbook of Virtual Environment Technology*, chapter 8, Motion Tracking Requirements and Technologies. 2002.

[5] Eric Foxlin, Yury Altshuler, Leonid Naimark, and Mike Harrington. Flighttracker: A novel optical/inertial tracker for cockpit enhanced vision. In *IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR 2004)*, 2004.

[6] Eric Foxlin and Leonid Naimark. Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker. In *Joint International Immersive Projection Technologies (IPT)/Eurographics Workshop on Virtual Environments (EGVE) 2003 Workshop*, 2002.

[7] C. Geiger, B. Kleinnjohann, C. Reimann, and D. Stichling. Mobile ar4all. In *Proceedings of the IEEE and ACM International Symposium on Augmented Reality (ISAR)*, 2001.

[8] Christian Geiger, Volker Paelke, and Christian Reimann. Mobile entertainment computing. In *TIDSE*, pages 142–147, 2004.

[9] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *proceedings of IEEE International Workshop on Augmented Reality*, pages 125–133, 1999.

[10] Dieter Koller. *Detektion, Verfolgung und Klassifikation bewegter Objekte in monokularen Bildfolgen am Beispiel von Strassenverkehrsszenen.* Infix, 1992.

[11] M. Moehring, C. Lessig, and O. Bimber. Optical tracking and video see-through ar on consumer cell phones. In *proceedings of Workshop on Virtual and Augmented Reality of the GI-Fachgruppe AR/VR*, pages 193–204, 2004.

[12] Wikipedia, the free encyclopedia. YUV. http://en.wikipedia.org/wiki/YUV.

[13] Iain E. G. Richardson. *Video Codec Design: Developing Image and Video Compression Systems.* Wiley, June 2002.

[14] J. Shen and S. Castan. An optimal linear operator for step edge detection. *Computer Vision, Graphics and Image Processing*, (54), 1992.

[15] Gilles Simon, Andrew W. Fitzgibbon, and Andrew Zisserman. Markerless tracking using planar structures in the scene. In Proc. of ISAR.

[16] Sargur N. Srihari. Analysis of textual images using the hough transformation. Technical report, State University of New York at Buffalo, 1988.

[17] Dean Wormell and Eric Foxlin. Vis-tracker: A wearable vision-inertial self-tracker. In *Joint International Immersive Projection Technologies (IPT)/Eurographics Workshop on Virtual Environments (EGVE) 2003 Workshop*, 2003.

Context Modeling for Device- and Location-Aware Mobile Web Applications

Michael Hinz Zoltán Fiala
Dresden University of Technology, Department of Computer Science
Heinz-Nixdorf Endowed Chair for Multimedia Technology
D-01062, Germany, +49-351-463-38516
{michael.hinz, zoltan.fiala}@inf.tu-dresden.de

ABSTRACT
Ubiquitous Web systems have to deal with varying context information in order to support context awareness. Accomplishing this requirement necessitates gathering, processing and representing that information, so that it can be used for adapting Web applications. In this paper we propose a mechanism for modeling dynamically changing context information like user's device capabilities and location. Furthermore we show how this modeling mechanism can be integrated into context-aware system architectures. With the aid of this approach we are able to perform a device- and location-aware online document generation of Web documents for mobile appliances.

Categories and Subject Descriptors
H.5.4 [**Hypertext/Hypermedia**]: Architectures. H.3.4 [**Systems and Software**]: User profiles and alert services.

General Terms
Management, Measurement, Design.

Keywords
Context modeling, context-awareness, device independence, ubiquitous computing, location-based services

1. INTRODUCTION
Today's WWW emerges to a medium of communication and cooperation. A multiplicity of users with different requirements uses the Web as a ubiquitous information store. In this way several trends such as personalization [1] and device independence [2] have raised that require the adjustment of Web applications to the user and its client device. Furthermore, with the emergence of location-based services, the client location becomes important context information, too. Still, today's location-based services (e.g. navigation system, location-aware museum guides) are fast-paced but usually stand alone applications not suited for the common Web. Furthermore their development is primarily driven by the mobile telecommunication industry and is therefore limited to applications on mobile phones. We claim that the main reason for this shortcoming is the still lacking support for automatically gathering and representing device and location context data. By improving the access to such context information the richness of communication in human-computer interaction increases, and makes it possible to produce more useful Web services [6]. To meet this challenge, this paper introduces context modeling mechanisms which enable the development of personalized device- and location-aware Web applications at the same time.

The paper is structured as follows. After addressing general context modeling aspects (section 2), modeling mechanisms for device capabilities (section 2.1) and the user's location (section 2.2) are illustrated. Beyond that, methods and techniques for sensing and representing that context information are presented. The described mechanisms were successfully integrated into a Web system performing a dynamic generation of context-aware Web pages (section 3).

2. CONTEXT MODELING
In the literature several definitions of the terms context and context-awareness can be found (e.g. [3], [4]). However, adding context to Web applications requires on the one hand the acquiring and modeling of context information such as location [5] as well as device context [2]. On the other hand that information has to be stored in a standardized representation. Without this application developers are left to develop ad hoc and limited schemes for storing and updating context information. However, the evolution of more sophisticated representations enabling a wider range of capabilities and a true separation of sensing context from the programmable reaction to that context is still an unsolved challenge [6]. To ensure this, the proposed modeling mechanisms represent the gathered information in a profile-based extensible context model ([14]) which relies on the W3C standard CC/PP [7]. Modeling components (e.g. for device, location and user modeling) guarantee a permanent monitoring of context and updating of the corresponding context profiles.

2.1 Device Modeling
Because of the requirements of mobile phones and resource restricted handheld computers their in- and output interfaces are limited e.g. in display size or number of buttons. Therefore they are far away from suitable interfaces to the Internet. Overcoming this burden requires the creation and publication of content that is tailored to users' platform capabilities and dynamically changing device properties. Therefore device capabilities have to be acquired, represented and provided to the document generation engine in the context aware system. In the next chapters those aspects will be explained more detailed.

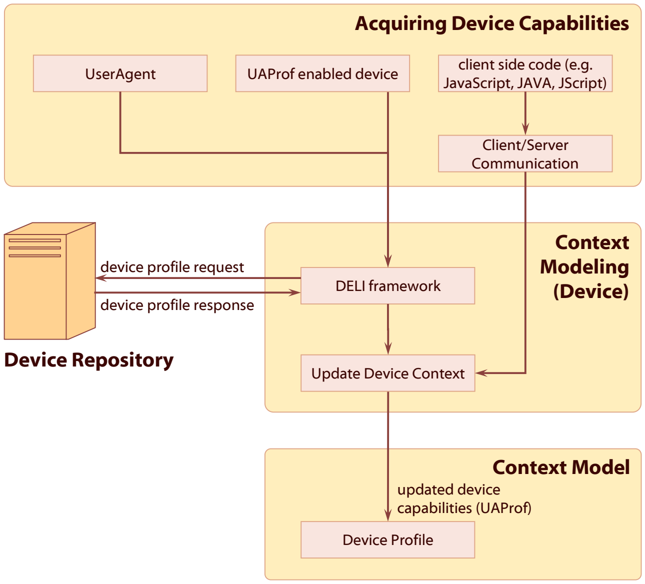


Figure 1: Modeling device capabilities

2.1.1 *Acquiring Device Capabilities*

Several strategies exist to acquire device capabilities for device sensitive Web applications. The most popular method is to analyze the HTTP user-agent parameter that comes with the HTTP request and map this parameter to a device or browser repository on the server side. However, this works only for nearly static device properties. Using the User Agent Profile specification (UAProf [8]) basing on the CC/PP framework [7] establishes a more effective mechanism for gathering dynamically changing device properties on the server by analyzing the UAProf enabled request. However, this specification only provides a common vocabulary for WAP devices. But most of the vocabulary can be adopted for other non WAP devices like normal Web browsers on desktop computers, notebooks or PDAs. In our work we extended the vocabulary in order to support those device classes. Further on for those device classes we provide a mechanism to transfer the gathered device capabilities within the HTTP request to the server.

In this way our device modeling mechanism illustrated in Figure 1 distinguishes between UAProf enabled devices, devices providing the user agent and devices giving support for client side code fragments like JavaScript, Jscript and Java (combinations are possible). Those client side code fragments are included during the Web document generation on the server and directly gather device properties on the client. The gathered information is encoded in a UAProf like representation and integrated in the HTTP request by a client/server communication component for processing that information on the server.

2.1.2 *Processing and Representing Device Context*

According to the gathered capabilities from the client, the server processes the corresponding device context. The processed context is represented as the device profile in the context model (see Figure 1 and Figure 3). The representation is based on the above mentioned extended UAProf format. The processing of the device context on the server depends on the obtained request.

1. If the request only includes the user-agent parameter this parameter is mapped to the according device profile in a device repository. Note that by using only this mechanism dynamically changing device properties (e.g. bandwidth or size of the browsers window) can not be taken into account.

2. If a UAProf enabled device sends a user-agent profile or a difference profile within the request, that information is handled by DELI [9] on the server side which provides an API for Java servlets to determine client capabilities using CC/PP and UAProf. The output of the DELI component makes a profile representing UAProf properties available.

3. Whereas today nearly only WAP 2.0 devices support UAProf, our system is also able to autonomously collect the devices properties of other end devices (e.g. Notebook, PDA) via the above mentioned client side code. The on the client gathered properties are sent within the HTTP request. Our server processes that information and merges it with an existing or by DELI generated device profile.

Together these mechanisms enable to acquire permanently changing device properties. The result is an always up-to-date device profile of the context model on which the document generation is based on.

2.2 Location Modeling

The trends of offering wireless networks and mobile internet access for a wide variety of mobile devices establish new areas of Web applications taking the location of the user into account. Since the location is important context information that changes whenever the user moves, reliable mechanisms for location-tracking are needed. Since normally all contents of a hypermedia Web application are delivered by Web servers, the location information is needed by the server which adjusts the contents according to the user's location. Generally there are two different strategies for obtaining the user location. It can be tracked by processing location data on a server (e.g. third party location server like Openwave Location Manager [13]). Another strategy is to obtain location information from a device-based application and transfer it to the server. Our concept for modeling the location of a client device supports both strategies. Figure 2 shows both scenarios using a location server abstracting from complex locating hardware for obtaining the client location.

2.2.1 *Locating Methods*

Locating methods can be categorized according to many aspects. Today many mobile devices (PDAs and Notebooks) use *satellite supported technologies* like GPS modules, which are either built in or available as extensions. Other devices like cellular phones (GSM and UMTS) or devices powered by radio networks (WLAN) are always connected to a base station handling a cell for mobile devices. So another technology for sensing the location is the use of *network supported cell information*. Generally it strongly depends on the application category which type of location processing should be used. E.g. because of the low signal intensity the conventional GPS does not work *indoors* (e.g. for museum guides or exhibition navigators) and is therefore restricted to *outdoor applications* [6]. Other aspects for a location method categorization are the supported dimension, accuracy, reliability and the positioning model of the method.

This shows that supporting location-based services in Web applications that are not limited to only one location method strongly requires the use of standard location protocols. Therefore in our concept for modeling the user location we use the Mobile Location Protocol (MLP [10]) standardized by the Open Mobile Alliance (OMA). This protocol enables a communication between location-based services and location infrastructure by means of XML interfaces. Figure 2 shows our architecture concept for modeling location information and storing that in the environment profile of our context model.

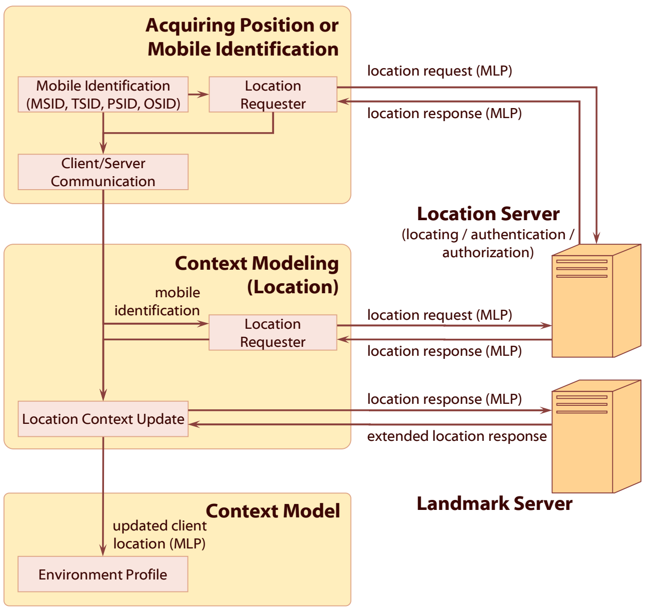


Figure 2: Modeling location and location context

2.2.2 Acquiring location information

As one possibility the architecture supports the acquisition of user location on client devices supporting the J2ME location API (JSR 179 [11]). This location API defines an optional package (javax.microedition.location) that enables developers to write wireless location-based applications and services for resource limited devices such as mobile phones, and can be implemented for any common location method [12]. Because the API implementation works on the client device the obtained location information has to transfer to the server for a server based generation of location-aware Web pages. This can be done by integrating that information in the HTTP request through the client/server communication component.

Another possibility to gather location information on the client is to receive the position of the user from a location server which analyzes network supported cell information according to the client mobile identification. In our proposed architecture (Figure 2) a location requester generates an MLP-request containing the mobile identification either in form of a North American Mobile Identification Number (MIN) or a GSM Mobile Subscriber ISDN (MSISDN) which are simply the mobile telephone numbers of the mobile subscriber. The following code snippet shows an example of a possible MLP-request generated by the location requester:

```
<svc_init ver="3.0.0">
  <hdr ver="3.0.0">
    <client><id>…</id><pwd>… </pwd></client>
  </hdr>
  <slir ver="3.0.0">
    <msids><msid type="MSISDN">1234567</msid></msids>
    …
  </slir>
</svc_init>
```

The MLP-response of the location server (e.g. OpenWave [13]) to the client could look like the following XML representation:

```
<svc_result ver="3.0.0">
  <slia ver="3.0.0">
    <pos>
      <msid type="MIN">3035551003</msid>
      <pd>
        <time utc_off="+0000">20050108014000</time>
        <shape>
          <CircularArea>
            <coord>
              <X>39 51 14.399N</X>
              <Y>105 02 53.858W</Y>
              <Z>5280</Z>
            </coord>
            <radius>1000</radius>
          </CircularArea>
        </shape>
        <speed>42.42640687119285</speed>
        <direction>45.0</direction>
      </pd>
    </pos>
  </slia>
</svc_result>
```

The determined position information is included in the next request to our server, which can use this information for generating location-aware Web documents. Again, this is done with the same mechanism of the client/server communication component that was already mentioned in Section 2.1.1.

Moreover the designed architecture supports the location acquisition on the server with a similar mechanism. In this case the client transmits its mobile identification directly to the server, which is responsible for the communication with the location server itself.

2.2.3 Processing and Representing Location

The location representations gathered either from the client or the server are stored in the environment profile of our context model. However, the location is represented by coordinate system-based position information which is difficult to use for adaptation processes because the author of a Web page who has to specify the adaptation rules is normally not familiar with it.

The solution is a landmark server storing landmarks which are locations associated with names and descriptions. The granularity of the landmarks depends on the Web application type. An outdoor route planner application for instance has the granularity of addresses (streets, house numbers) whereas an indoor application like a museum guide has a finer granularity.

The landmark server maps the location contained in the MLP-response to associated names and descriptions of that location. By grouping of locations different location granularities are taken into account. After the mapping the extended response is used to update the environment profile of the context model (Figure 2).

For that purpose the MLP extension mechanism is used. The following example shows how the response is extended:

```
<svc_result ver="3.0.0">
  <slia ver="3.0.0">
    <pos>…</pos>
    <LocationExtension>
      <LocationMark>
        <Group>MMTChair</Group>
        <Name>Room 202</Name>
        <Description>Projekt Amacont</Description>
      </LocationMark>
    </LocationExtension>
  </slia>
</svc_result>
```

The extended MLP-data is represented in the context model and can be used for performing adaptation processes.

3. INTEGRATION INTO A CONTEXT AWARE SYSTEM ARCHITECTURE

The proposed context modeling mechanism assures an always up-to-date context model containing information about the user's device and location. Furthermore the context model includes additional information about the user which can be used for personalization aspects. For a detailed introduction in those aspects and user modeling process the reader is referred to [14].

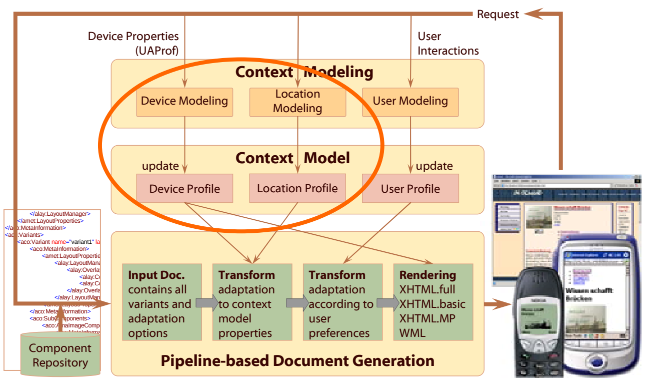


Figure 3: Context aware system architecture

Figure 3 shows how the proposed context modeling mechanisms can be integrated in an architecture for generation of context-aware Web pages. Note that the device and location modeling mechanisms (see Section 2) are located in the highlighted architecture components (compare Figure 1 and Figure 2). For each user request, a complex document is retrieved from a component repository. According to the context model, this document is adapted to user, device and location properties and rendered to a specific output format (XHTML, cHTML, WML etc.). Proving feasibility that overall architecture was implemented on the basis of the XML publishing framework Cocoon.

4. CONCLUSION AND FUTURE WORK

The work we proposed enables Web system architectures to generate context-aware Web pages adapted to users' device capabilities and location information. Therefore we presented a mechanism for modeling that dynamically changing context information and show how it can be integrated in a context-aware system architecture. Future work will concentrate on distributing the context model to enable the user to decide which of the gathered information she/he is willing to share with the server architecture. Another aim is the performance enhancement of the proposed architecture since generating context-aware Web documents makes high demands on online Web publishing systems.

5. REFERENCES

[1] Fink, J., Kobsa, A. *A Review and Analysis of Commercial User Modeling Servers for Personalization on the World Wide Web*. User Modeling and User-Adapted Interaction 10 (2–3), 2000, pp. 209-249.

[2] Butler, M., Giannetti, F., Gimson, R., Wiley, T. *Device Independence and the Web*. IEEE Internet Computing, Vol. 6, No. 5, Sep.-Oct. 2002, pp. 81-86.

[3] Dey, K., Abowd, G. D. *Towards a better understanding of context and context-awareness*. In Workshop on “The What, Who, Where, When, and How of Context-Awareness”, Conference on Human Factors in Computer Systems (CHI2000), 2000.

[4] Dey, A. K. *Understanding and using context*. Personal and Ubiquitous Computing, Special issue on Situated Interaction and Ubiquitous Computing 5, 1, 2001.

[5] Abowd, G., Mynatt, E. *Charting past, present and future research in ubiquitous computing*. ACM Transactions on HCI, Special issue on HCI in the new Millenium, 7(1), March 2000, pp. 29-58.

[6] Abowd, G., Mynatt, E., Rodden, T. *The Human Experience*. IEEE Pervasive Computing Magazine, 1(1), January-March 2002.

[7] Klyne, G., Reynolds, F., Woodrow, C., Ohto, H., Hjelm, J., Butler, M., Tran, L. *Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0*. W3C Recommendation, January 2004.

[8] Wireless Application Group. *User Agent Profile Specification*. Open Mobile Alliance WAP Forum, 2001.

[9] Butler, M. *DELI: A DElivery context LIbrary for CC/PP and UAProf*. HP, External Technical Report HPL-2001-260 (revised version 02/08/2002), 2002.

[10] Open Mobile Alliance. *Mobile Location Protocol (MLP)*. Enabler Release Definition for Mobile Location Protocol (MLP) Candidate Version 3.1, March 2004.

[11] *Location API for J2ME (JSR 179)*. Java Specification Request, September 2003.

[12] Mahmoud, Q. *J2ME and Location-Based Services*. Technical Articles of the Sun Developer Network, March 2004.

[13] OpenWave. Location Studio SDK http://developer.openwave.com/dvl/tools_and_sdk/openwave_mobile_sdk/location_studio/

[14] Hinz, M., Fiala, Z., Wehner, F. *Personalization-based Optimization of Web Interfaces for Mobile Devices*. In Proceedings of the MobileHCI 2004, Glasgow, Scotland, September 2004.

Ubiquitous Computing for the Public

Patrik Osbakk
Jayway AB
Box 571, 201 25
Malmö, Sweden
patrik.osbakk@jayway.se

Erik Rydgren
Jayway AB
Box 571, 201 25
Malmö, Sweden
erik.rydgren@jayway.se

ABSTRACT
Despite the progress made in Ubiquitous Computing since it was first envisaged back in 1987 [1], its use is still isolated to the research community. In this paper, we present why we feel the use of off-the-shelf mobile information devices may aid in bringing ubiquitous computing to the public. We also outline an ongoing project that we are undertaking in hope to introduce ubiquitous computing to a wider audience together with some preliminary results.

Keywords
Ubiquitous Computing, Mobile Devices, Bluetooth

1. INTRODUCTION
Has the progress made over the last decade brought us closer to being able to achieve the vision of ubiquitous computing? Yes, most definitely. Much experience has been gained from the many ubiquitous computing projects that have been performed world-wide, covering such topics as location sensing [2], context-awareness [3], privacy [4], fieldwork [5], health care [6], etc. Even though there is still a lot more to do within all of these areas and many more, the field as a whole has matured and perhaps most importantly it has been widely recognised and established. Significant progress has also been made with the underlying technologies. Today inch-scale computing devices are readily available to the masses. These come with various types of wireless connectivity (gprs, bluetooth, wifi), have decent battery life, and most importantly can be bought at reasonable prices.

However, despite the progress made, the benefits of ubiquitous computing are still largely restricted to the research community. Few applications and systems find their way into the hands of the general public, unless they are part of an organised study. This is regretful and will increasingly become a problem if the vision of ubiquitous computing is to be achieved. It should not be forgotten that the overall aim is *not* the development of novel applications nor is it to simply distribute thousands of invisible and interconnected devices into our environment. The aim is to aid people in their daily lives. After all the vision was born when attempting to reduce the complexity of the personal computer and the attention it requires [1]. As Mark Weiser so elegantly put it “The most profound technologies are those that disappear. They weave themselves into the fabric of our everyday life” [7]. Thus it is not enough that small devices can be manufactured nor that great applications exist. For the aim to be achieved the technology needs to be gradually introduced into our everyday life so that people get the opportunity to learn and experience it. First then will it fade into the background and be ubiquitous.

2. MOBILE DEVICES
But before people are able to start using ubiquitous computing technology it must first of all be available to them. This may appear to be a trivial step but it is a step that needs to be taken nonetheless. Ubiquitous computing artefacts should be possible to obtain in the same manor as any other consumer product. It cannot be expected that the general public will search specialist forums nor to go through any other complicated procedure to obtain the technology. The technology must also be provided in a form people understand and can use, otherwise it does not matter if it is readily available. Thus the technology needs to be self-explanatory and the interaction so natural that it can occur without instructions.


Figure 1. Mobile devices

How then can we make ubiquitous computing available to the public? Well, even though most people may not think about it, they interact with inch-scale devices on a daily basis. The use of devices such as mobile phones and Personal Digital Assistants (PDA) is today so widespread in society that they are becoming pervasive. Increasingly these devices also come with the capabilities required for them to be suitable as a platform for ubiquitous computing, something several projects have shown [8][9].

Our position is therefore that off-the-shelf mobile devices should be used whenever possible. This is also the approach we are taking in our efforts to bring ubiquitous computing to a wider audience. The main benefit as we see it is that this will enable ubiquitous computing artefacts to be made available at a much larger scale. Mobile phones and PDAs are readily available to consumers and there is also a growing market from which software can be obtained. Another benefit is that the public already has come into contact with these devices. This will hopefully make it easier for users to start using the artefact produced and to let them see beyond the technology itself.

3. THE BLUE REMINDER
In an attempt to introduce ubiquitous computing to a wider audience we are currently undertaking a project which uses today’s pervasive technology to tackle the problem of remembering things.

It is no secret that we all tend to forget things from time to time. To minimise our lapses in memory we can use various tools in our daily lives. The post-it note is one such tool. We may for example stick a post-it note on a book with the date when it is due back at the library. The note will then be able to prompt us, when seen, that there is something we need to remember and what it is. Although this traditional technique works well, it is not suitable for all situations. For example we may want to discuss something in person with a colleague the next time we see them. In this case we would like to be reminded only when the colleague is around. The use of an ordinary post-it note in this situation is far from ideal. The best we can do is to attach the note to something we carry with us and hope that we see it at the right time. If we forget to look at our notes, which is bound to happen occasionally, then we may miss those times when we really need to get reminded.

Figure 2. Post-it notes

Is there then an alternative? We believe there is. By using virtual post-it notes this problem can be overcome. Virtual notes are far more flexible than the traditional paper notes we otherwise use. They can be attached invisibly to anything with a unique identifier. This can include everything from people to places and things [10]. The idea is that when a person thinks they need to be reminded about something they will write a virtual post-it note and attach it to a suitable identifier. In the previous example this would be an identifier representing a colleague. Then the next time the presence of the colleague is detected the virtual note can be displayed to remind the user. The use of virtual notes also has the advantage that they allow other factors to be considered as well. For example they can be set to only appear on certain days such as birthdays or anniversaries, when a certain group of people meets, at a particular time of the day, etc.

The BlueReminder is a system being developed on this concept. Through the use of mobile phones we provide users with the ability to both write and read virtual post-it notes. In the system notes are written with a subject, a message, and a triggering condition. This condition can be the presence of one, a group, or one in a group of identifiers at any or a specific date. In addition to this the user has the ability to customise options such as the validity period and the notification action. The system will then at predefined intervals check if any of the written notes should be triggered. If a match occurs the user is notified and can view their note to be reminded.

4. Architecture

To realise the project we have chosen to use standard off-the-shelf mobile phones. They provide us with a uniform platform for which the application can be written as well as the necessary capabilities to detect nearby identifiers. The former comes in the shape of the Java 2 Platform, Micro Edition (J2ME) [11]. Whilst not the most efficient platform, it does provide the widest possible support. The J2ME technologies used include the Compact Limited Device Configuration (CLDC) 1.0 [12] and the Mobile Information Device Profile (MIDP) 2.0 [13]. The latter, i.e. detecting nearby identifiers, is achieved through the use of Bluetooth technology. First of all we assume that people etc. are identified by Bluetooth tags, e.g. mobile phones. Then this assumption is used to allow their presence to be detected using the Bluetooth Service Discovery Protocol. Hence it should be noted that no data is transmitted between devices with the BlueReminder application. The only data used are visible Bluetooth identifiers. This ability to use the mobile phone's Bluetooth functionality is provided by the JSR-82 API [14], which is implemented in several currently available devices.

Device discovery

The JSR-82 API provides a DiscoveryAgent class that can be used to create a search agent object. By using this object, an inquiry for remote Bluetooth devices can be initiated on the local device. The agent is tied to a listener object and reports events such as the discovery of a device and the completion of a search session back to it. The listener object in turn implements methods for handling each of these events.

In the BlueReminder, the search functionality is located in a separate thread that periodically calls the start method of the search agent, waits for it to notify the completion of the search, and finally sleeps for a set period of time. The time taken by an inquiry varies with the number of surrounding devices and the distance to them. Usually it takes about 15 second though. After each search, the list of discovered devices is then matched against the contacts of each available note. If a match occurs the an alert box is displayed together with an optional notification. The notification can be a combination of a vocal signal, a vibration, or flashing lights. On some devices where the screen is not available to an application at all times, for example when the lid on a foldable phone is closed, the BlueReminder periodically request access to the screen once a note has been triggered. This ensures that the alert box is displayed once the screen is made available.

Searches can also be initiated manually to add new contacts, the results are then viewed as a list on the screen. The devices presented are either marked as known or unknown depending on whether they are used in the contacts list or not. By clicking on a device identifier in the list it can be added as a new contact, or to an already existing contact. Thus each contact can have more than one device id. This is important as it is expected that people may carry more than one Bluetooth enabled device or switch between several.

Data storage

The persistent data storage available on the J2ME platform differs from that on traditional computing platforms. While the most obvious difference might be its limited size, the important difference is that applications do not enjoy direct access to the file system. Instead the data storage functionality is provided in the form of record stores. An application can have several record stores, each identified by a string value. These record stores then contain data in the form of records, each identified by an integer value.

The types of data that need to be stored by the BlueReminder are notes, contacts, and settings. For this three separate record stores are used. To simplify the handling of data, a common mechanism has been created that exploits the similarities of the different record stores. The mechanism consists of a collection of general storage classes and interfaces. For each type of record store the methods needed to convert the relevant objects into byte streams has then been implemented. This is required since a record holds data in the form of byte arrays.

The BlueReminder also implements non-persistent storage structures to be used while the application is running. One such structure is the device buffer. The buffer is used to store an adjustable number of devices found in past searches using a first in, first out buffering scheme. This is an important functionality as it supplies the users with a way of adding discovered devices to the contacts list at a time they find convenient rather than upon discovery.

Graphical user interface

Two levels of GUI design are offered by MIDP2.0. A lower level API makes it possible for a developer to assume complete control over the appearance of an application. This is however a very timeconsuming way of developing, thus J2ME also provides a higher level API. At the higher level, classes implementing common objects such as lists, buttons and textboxes are provided. For the BlueReminder the higher level API was considered more appropriate since the application only utilises standard GUI components. This also has the added advantage that the layout of the application's interface is optimised for the devices on which it is run, lowering the learning curve for new users.

Each screen of the application has been implemented as its own class. Whilst not the most efficient design it does make the code highly readable and easier to extend. The GUI receives input from the user through the device keyboard or through the touch screen if this feature is supported. The input is directed to the GUI component currently in focus, such as a text field. Each screen class implements a command listener interface. The method of this interface is called when the user submits a command through clicking a button, selecting a menu alternative or clicking an item in a list.

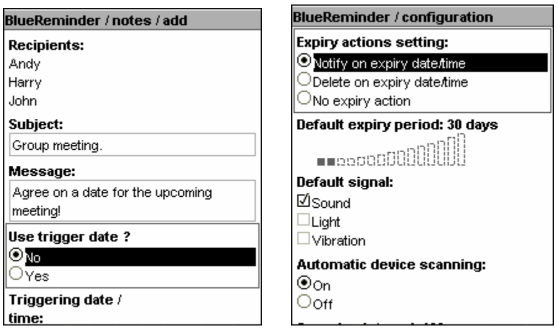


Figure 3. BlueReminder screenshots

5. Preliminary findings

The application has been tested on two popular types of mobile phones from two well known manufacturers. The first phone is a high-end device with a large touch-screen while the second phone is a medium-range device with a small screen supporting output only. The former runs the Symbian operating system and the latter run a vendor specific. The aim of these initial tests has been to verify the concept as well as the conditions for wide scale deployment.

Concept

Overall the concept of the BlueReminder has proved to work well. Notes can be written and reminders issued in everyday situations. This certainly extends the usefulness of these phones. Even though the appearance of the application differs due to the different nature of the devices, the application has been found to be easy to use on both phones. The use of the higher level API and its device specific implementation certainly plays a part in this. This is an important result to enable wide scale deployment as otherwise the effort needed to port the application to each device would be significant.

Energy consumption

We have also investigated how the BlueReminder affect the energy consumption of the mobile device. This is important as the energy consumption need to be kept to a minimum as not to influence the users' desire and ability to run the application. The approach used has been to leave a number of mobile phones running for eight consecutive hours in three different states:

- State 1: Phone in standby mode, Bluetooth switched off
- State 2: Phone in standby mode, Bluetooth switched on.
- State 3: Phone in standby mode, Bluetooth switched on, BlueReminder running

At the start of the test all the mobile phones were fully charged and the power level recorded. Then at the end of the test the power level was checked and recorded again for each phone. By comparing the results between the different states for a given model an indication of the energy consumption has been found. The preliminary results show that running the BlueReminder consumes a significant amount of battery power. This was expected as the regular device discovery operations actively use the Bluetooth radio. Initial figures indicate that state 3 consume twice the energy as state 2. However the results also show that whilst the energy consumption is significant it does not drain the batteries to the degree as to hinder everyday operation. After the BlueReminder has run for 8 hours (state 3) there is still approximately 90-95% of a full battery left. These figures are very much preliminary, further tests need to be performed to verify them.

Stability

Finally tests have been performed to examine the stability of the BlueReminder. Stability is here defined a measure of the predictable behaviour the application exhibits when run over long periods of time. For the benefits of the BlueReminder to be gained it needs to run continuously in the background of the device, routinely performing a device discover. If the application is terminated, the user runs the risk of not being reminded. Good stability is therefore a primary design goal. Throughout the design process the minimum period of time the application must be stable was set to 24 hours, meaning that the application must at the very least remain stable for this period under normal conditions in order to be practical.

During our testing stability problems have been discovered, on the more powerful of the two test platforms. After running the application for several hours, sometimes several days, the Bluetooth service of the device unexpectedly stalls. This prevents the BlueReminder from performing anymore searches. Whilst this abnormal state can be detected by measuring the time an inquiry takes, the J2ME platform does not provide any mechanism to address this issue. The only way to recover from the abnormal state is by manually restarting the Bluetooth service on the device through the control panel of the phone in question. Thus the application has been adapted to alert the users if the Bluetooth service stalls. This is not optimal though.

Two interesting observations have been made regarding the probability of a lockup occurring. First, the probability for the Bluetooth to stall is higher on some devices than others, even if they are of the same model, use the same firmware, and run identical versions of the application. This leads us to believe that there are other factors involved. We can only speculate what these factors are but possible causes include different patterns of device usage or variations in the quality of the hardware. Secondly, the time before a device enters into an abnormal state becomes significantly shorter as searches are performed more frequently. A sleep period between searches of at least 180 seconds was found to reduce the probability of service lockups significantly on most of the devices tested.

To establish where the problem lays, a simplified test application has been developed in Personal Java that performs an equivalent device search. This implementation has them been subjected to the same type of stability tests as the J2ME implementation. The results found were the same. After different lengths of time the Bluetooth service stalls on the mobile devices. This indicates that the problem is neither in JSR-82 API nor in the J2ME implementation of the BlueReminder. The fact that we have so far only observed this problem on one of the platforms further strengthens our belief that it stems from either the low-level operating system interface towards the hardware or the actual hardware itself.

On the high-end devices infrequent virtual machine errors have also been encountered. By careful optimisation of the application's memory usage, these problems have been minimised. On the second, medium-end, devices no stability problems occurred per say but the application times-out after being inactive for a longer period of time. The reason for this is that the device is not capable of running the application in the background. This has obviously limited the ability to perform long running test on this device. However the tests performed have not given any indications of stability issues being present.

Overall the stability tests suggest that there is a need to look at the underlying operating system and the Java Virtual Machine. However the fact that one of our two test platforms did not present any stability problems indicates that BlueReminder itself is stable. Also the relatively low frequency of problems suggests that it is possible to use the application on a day to day basis. Finally upcoming mobiles are likely to improve upon stability as both the JVM and the Bluetooth services matures.

6. FURTHER WORK

Our findings have identified several areas for further work.

The current test results need to be further verified. This includes running more test on a wider range of devices. Whilst the results are not expected to differ, it will allow them to be quantified. In particular it would be useful to gather statistical information on how long the application is stable as well as more exact measure of the energy consumption.

There are a number of additional functions, minor changes and optimisations to be made before a large scale deployment can be made. Currently the BlueReminder lacks the ability to import/export contacts and notes. This functionality is needed for corporate use. Then there is the issue of the stability. This issue need to be further investigated, ideally in close cooperation with device manufacturers, to see if the situation can be improved upon. Finally there is the energy consumption for which the current implementation has not been optimised. Here improvements can be made by for example: only searching when there are notes, increasing the interval between searches, disabling the search at certain times, etc.

It is also desirable to run a case study with the target audience, the public, to study their use of the technology. Interesting topics include their interaction with the devices, how they incorporate the new tool in their daily lives, how they consider their privacy being affected, etc.

7. RELATED WORK

The idea of an virtual post-it note is not new [15][16]. The stick-e notes concept allowed electronic notes to be attached to locations. These notes would then trigger events upon being detected, which in general involved displaying note information to the user. The stick-e note concept has also been proven to work in a fieldwork environment. This is a good indicator that verifies that the concept indeed can be used in real situations. The feasibility of using bluetooth on mobile devices to mediate contextual information has also been shown [17]. MobiTip for example uses bluetooth to communicate people's comments on things in the environment from user to user. Collectively these comments then form the basis for tips that can be given to users.

8. CONCLUSION

Introducing ubiquitous computing to the general public is not an easy task. Making it a natural part of everyday life is even more difficult. However a first step towards achieving this goal will be to make the technology available in a form people understand and can use. Our position is that the use of off-the-shelf mobile devices can aid us in taking this step. Our ongoing project, the BlueReminder, uses today's mobile phones to provide a pervasive service that allows virtual post-it notes to remind users. What makes this project different from previous efforts is that the focus is on aiding people's interaction *with each other*. Furthermore the project is undertaken with the explicit intention of *reaching a wider audience*. Our results so far highlight some issues that need to be investigated further, in particular with respect to stability over long periods of time. However despite this the overall indication is that a large scale deployment of ubiquitous computing technology is indeed within reach today.

9. ACKNOWLEDGEMENTS

Many thanks to the joint Jayway/Leadway project team for their effort in realising the BlueReminder.

10. REFERENCES

[1] Weiser, M., R. Gold, et al. (1999). "The origins of ubiquitous computing research at PARC in the late 1980s." IBM Systems Journal 38(4): 693-696.

[2] Harter, A., A. Hopper, et al. (2002). "The Anatomy of Context-Aware Applications." Wireless Networks 8(2-3)

[3] Schilit, B. N., D. M. Hilbert, et al. (2002). "Context-Aware Communication." IEEE Wireless Communication 9(5)

[4] Osbakk, P. and N. Ryan (2004). The development of a privacy-enhancing infrastructure: Some interesting findings. Ubicomp Privacy: Current Status and Future Directions Workshop, UbiComp 2004, Nottingham, UK.

[5] Ryan, N., J. Pascoe, et al. (1999). FieldNote: a handheld information system for the field. TeleGeo'99, 1st Int. Workshop on TeleGeoProcessing, Lyon, France.

[6] Consolvo, S., P. Roessler, et al. (2004). The CareNet Display: Lessons Learned from an In Home Evaluation of an Ambient Display. UbiComp 2004, Nottingham, UK.

[7] Weiser, M. (2002). "The Computer for the 21st Century." Pervasive Computing 1(1): 19-25.

[8] Rukzio, E., A. Schmidt, et al. (2004). An Analysis of the Usage of Mobile Phones for Personalized Interactions with Ubiquitous Public Displays. Workshop Ubiquitous Display Environments, UbiComp 2004, Nottingham, UK.

[9] Mäntyjärvi, J. and T. Seppänen (2003). "Adapting applications in mobile terminals using fuzzy context information." Interacting with Computers 15(4): 521-538.

[10] Kindberg, T., J. Barton, et al. (2002). "People, Places, Things: Web Presence for the Real World." Mobile Networks and Applications 7(5): 365-376.

[11] Java 2 Platform, Micro Edition. Last Accessed 26/04/2005. *http://java.sun.com/j2me/*

[12] JSR-000030 J2ME Connected, Limited Device Configuration. Last Accessed 26/04.2005. *http://jcp.org/aboutJava/communityprocess/final/jsr030/index.html*

[13] JSR-000118 Mobile Information Device Profile 2.0. Last Accessed 26/04/2005. *http://jcp.org/aboutJava/community process/final/jsr118/index.html*

[14] JSR-000082 JavaTM APIs for Bluetooth. Last Accessed 26/04/2005. *http://jcp.org/aboutJava/communityprocess/final/jsr082/index.html*

[15] Brown, P. J., J. D. Bovey, et al. (1997). "Context-aware applications: from the laboratory to the marketplace." IEEE Personal Communications 4(5).

[16] Pascoe, J., D. Morse, et al. (1998). "Developing Personal Technology for the Field." Personal Technologies 2(1)

[17] Rudström, Å., M. Svensson, et al. (2004). Using Bluetooth as a Mediator of Social Context. UbiComp 2004 Adjunct Proceedings (demo), Nottingham, UK.