

Identifying User Requirements for a Mobile Terminal Centric Ubiquitous Computing Architecture

Eija Kaasinen, Marketta Niemelä, Timo Tuomisto and Pasi Välikkynen
VTT, Technical Research Centre of Finland
Tampere, Finland
{{firstname.lastname}@vtt.fi

Vladimir Ermolov
Nokia Research Center
Helsinki, Finland
Vladimir.Ermolov@nokia.com

Abstract

System level solutions affect many properties of ubiquitous applications and thus also user experience. That is why user point of view should guide the design of mobile architectures although the users will see them indirectly, via the applications. This paper describes our approach in identifying user requirements for a ubiquitous computing architecture that facilitates mobile applications sensing their environment. The sensing is based on wireless connectivity to tags and sensors in the environment. We illustrated a representative set of future applications as scenarios and proof-of-concepts and evaluated them with potential users. Scenarios were analyzed to identify generic use cases and to understand the implications of the user feedback on the architecture. Our experiences show that user requirements for system level solutions can be identified with this approach. We identified several requirements for the architecture dealing with user interaction, wireless measurements, context-awareness, taking applications into use and ethical issues.

1. Introduction

Mobile terminals are increasingly evolving into tools to orient in the environment and to interact with it, thus introducing a mobile terminal centric approach to ubiquitous computing [1]. With ubiquitous computing, many properties of the applications are defined by the underlying system level solutions that are in charge of providing wireless connections to the objects in the environment, providing external connections, identifying contexts and so on. To ensure user acceptance of ubiquitous applications, we should be able to get user feedback already to the design of the system level solutions. Although human-centered

design of end-user applications is a well-established and even standardized process [2], approaches to studying systematically user requirements for system level solutions are still rare.

In this paper we describe our experiences in identifying user requirements for a ubiquitous computing architecture. The research work was carried out within an EC-funded research project MIMOSA, Microsystems Platform for Mobile Applications and Services [3]. The project is developing novel microsystems solutions for wireless sensors and tags as well as a mobile platform that facilitates connecting those microsystems wirelessly to mobile phones and other personal mobile devices.

We start this paper with an overview of related research; reported experiences in extending human-centered design approach to system level solutions. After that we introduce our design target: the microsystems platform for mobile services and applications. Then we describe the user requirements definition process that we applied in the MIMOSA project. The rest of the paper describes and analyses key user requirements identified and their implications on the architecture.

2. Related research

Implicit in the idea of ubiquitous computing are the notion of infrastructures – some new, some existing, some virtual, some physical, some technical, some social – all coming together in a seamless way [4]. The potentially vast collection of devices, sensors and personalized applications presents a remarkable challenge to the design of ubiquitous computing infrastructures [5]. Some researchers have touched the issue of focusing more on infrastructures although they do not report actual experiences [6] – [8]. Islam and Fayad [8] emphasize the necessity to develop devices parallel to mobile infrastructures: new types of contents require new types of devices; new network

options require devices that can choose the most suitable network in each context of use and so on.

The well-established human-centered design approach as defined by the ISO 13407 standard [2] is focused on the design of individual applications. The standard guides to iterative prototyping that is not easy in current complex and concurrent application development [9]. For system level solutions that include both hardware and software, iterative prototyping would be almost impossible. We will need new methodological approaches to identify user requirements for system level solutions.

Edwards et al. [10] have presented encouraging experimental results of two case studies where they have identified user requirements for software infrastructures: a file system and a context toolkit. Edwards et al. [10] emphasize the importance of simple proof-of-concepts that illustrate core infrastructure features. User evaluations of these proof-of-concepts gave early feedback to the design of the infrastructure and ensured that the most central system level features were designed to facilitate user-friendly applications.

The case study described in this paper has taken a new challenge as we are focused on a platform architecture that includes both hardware and software. The selection of possible services that our platform will facilitate is much broader than the applications dealt with in the study Edwards et al. [10]. That is why in our case we needed to complement proof-of-concepts with lighter ways to illustrate and evaluate future possibilities – scenarios as described in section 4.

3. The MIMOSA architecture

The planned MIMOSA platform architecture provides mobile users with a smooth transition from current mobile services to ubiquitous computing services. In MIMOSA approach the personal mobile terminal becomes a tool for the user to sense the environment and to interact with it. The approach is based on short-range connectivity that can be set up with relatively modest investments in the infrastructure. A wide range of consumer applications is covered as the architecture facilitates wireless connections to different tags and sensor units [3].

At the top level MIMOSA architecture can be divided into four physical entities and wireless interfaces between the entities. The entities are mobile terminal, remote application server, sensor radio node and wireless remote powered sensor (WRPS). The physical entities are illustrated in Figure 1. As the MIMOSA project is developing novel microsystems solutions

parallel to the development of the architecture, for instance the targeted reading distances are longer and the targeted component sizes smaller than in currently available tag and sensor solutions.

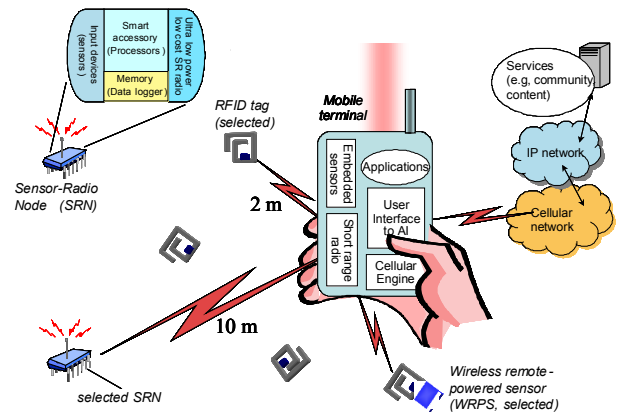


Figure 1. The MIMOSA architecture facilitates local connectivity to tags and sensors

The mobile terminal, the sensor radio node and WRPS all have local connectivity capability so that they can communicate with each other. The terminal can collect information from the sensors automatically or based on user initiation. The terminal is also capable to read different RFID (Radio Frequency Identification) tags. Optional remote connectivity allows for remote application servers in the Internet to apply locally obtained sensor data within the context of specific services and applications.

Terminal device is the trusted device of the user. Most important blocks of the terminal are processor platform, local connectivity module, user interface hardware and embedded sensors. Processor platform is running an operating system. User interface hardware supports easy and flexible interaction between user and terminal. Local connectivity module offers connections to surrounding sensor devices and tags. Individual applications handle information collection, processing and displaying of the results. Context engine is a software module that monitors sensor measurements, identifies context atoms and further contexts and informs applications about contexts according to how the applications have registered for contexts.

Wireless remote powered sensor (WRPS) has architecture similar to RFID tags. Parts of the WRPS are digital control logic, including radio protocols and information controlling, analogue front end, rectifier, power management, and sensor. From the system point of view WRPS can be seen as a special kind of tag from where can be read not only the tag ID but also a sensor reading.

Sensor radio node (SRN) provides a more versatile and longer range sensor connection. SRN consists of a micro-controller-based host (MCU), local connectivity module and sensors. Micro-controller has a processor core that enables application specific software running on the sensor unit. This includes communication protocol stack and sensor information processing. Pre-processing capability in the sensor radio node can provide high-level sensor information for user applications running on the terminal. Sensor radio node can use the radio to advertise services that are available.

As the design and implementation work of the architecture started at the same time as our user requirements definition process, we could not target to influence much on the first architecture prototype but the focus was on the next generation architecture that would be commercialized based on the results of the MIMOSA project.

In the following, we will use the term platform to describe the personal mobile terminal enhanced with local connectivity. Sensor unit is used as a term to describe generally the different sensors and sensor-radio nodes. Architecture describes the whole entity of the mobile terminal, sensors, sensor radio nodes, tags, remotely connected servers as well as related system level software.

4. The process of identifying user requirements

When identifying user requirements for system level solutions, it is essential to see the wide variety of future applications. We decided to focus on four promising consumer application fields: sports, fitness, health care and housing. To enhance coverage we were also studying general everyday applications.

MIMOSA project gathered together a multitechnical project group that included partners developing individual technical solutions and application field experts from different organizations. A scenario workshop was organized in the beginning of the project to define a common vision and to integrate the ideas of the partners. The common vision was illustrated in the form of totally 36 usage scenarios that described comprehensively different application possibilities facilitated by the architecture in the form of excerpts of usage situations.

The scenarios were evaluated with potential users and application field experts (94 people in total) in focus groups (3 to 8 participants in each group interview) and by individual questionnaires where the participants assessed the value of proposed solutions.

In addition we evaluated the scenarios with our project partners with web questionnaires.

Based on the scenario evaluation results, we refined the scenarios ending up with 20 more focused scenarios. Those second generation scenarios presented applications that were accepted by end users and application field experts. Also the credibility of the proposed technical solutions had been checked with project partners. The following excerpt of a golf scenario as well as Figure 2 illustrates the scenarios:



Figure 2. The golf scenario

Matthew is quite serious with his golf practice. As he arrives at the first tee at the golf course, his golf application on his wristop is activated. All of Matthew's golf clubs are equipped with tags and sensors. His wristop computer runs a golf application that helps him to keep track of the clubs he has used and the courses he has played at. During the game, the golf clubs send measurement data of each swing to the wristop computer. At home Matthew compares his results to those of Tiger Woods.

After the scenario refinement, we analyzed the scenarios into use cases and further into sequence diagrams to describe in details what kind of communication they indicated between the user, the application and the different parts of the architecture. The use cases and sequence diagrams were discussed through with the technology developers of the architecture to ensure that the proposed sequences were correctly interpreted and possible to realize [1]. Figure 3 illustrates one of the sequence diagrams related to the golf scenario.

The users could give feedback on many application features based on the illustrated scenarios but features where the look and feel was essential were difficult to imagine and comment. To study and compare alternative look and feel properties, we built proof-of-concept demonstrators. Those simple prototypes were

based on current technology and they illustrated basic application and interaction concepts.

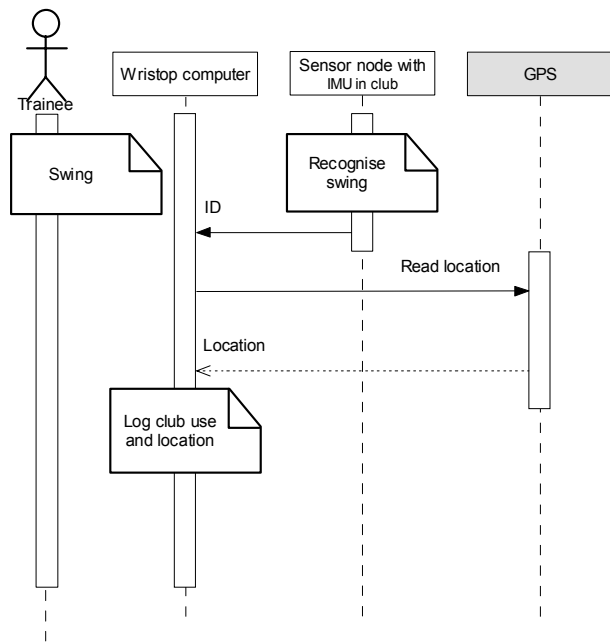


Figure 3. A sequence diagram of the analyzed golf scenario

The proof-of-concepts included for instance a PDA-based prototype that illustrated user interaction with tags in the environment, context-aware everyday services implemented on a tag-reader phone, golf proof-of-concept with motion-monitoring of the club and fitness proof-of-concept integrating different fitness measurements. We evaluated those proof-of-concepts each with 3-8 users in usability tests in laboratory conditions or in the field.

The user evaluations gave us feedback both on proposed functionalities and on the quality attributes for those functionalities. To identify the implications on the architecture, we analyzed the user cases and related user feedback together with architecture designers in several workshops. The emphasis was on generic use cases repeating from one scenario to another as they obviously had strong impact on the architecture [1].

When user requirements are defined for an individual application, we can get quite precise requirements. With system level solutions, the users are commenting individual applications and we need to integrate and interpret their feedback to see the implications on the architectural level. The applicable research methods are mainly qualitative and that is why identified requirements tend to be descriptions of trends rather than statistical evidence. We ended up

with a set of user requirements for the architecture classified as: 1) User interaction by physical selection, 2) Gathering measurement data from wireless sensors, 3) Context-awareness, 4) Taking applications into use and 5) Ethical issues. In the following section we will describe key user requirements identified and their implications on the architecture using this classification.

5. User requirements for the architecture

5.1 User Interaction by Physical Selection

Physical selection allows the user to select an object for further interaction. The selection can be done by touching or pointing with the mobile device but the device can also scan the environment and propose objects for interaction [11]-[14]. In our scenarios physical selection was a recurring usage pattern. The scenarios gave us an overview of the possible ways of user interaction but a user study was required to evaluate and compare the usage patterns more thoroughly and to identify parameters for the physical selection methods. We built a physical selection prototype, which emulated user interaction with MIMOSA style passive RFID tags and supported selection by touching, pointing and scanning. The user study with thirteen test subjects was then conducted using this prototype to find out answers to the specific research questions such as reading distances [15]. In the following, we first describe general user requirements regarding physical selection (gathered by scenario analysis) and then additional requirements for touching and pointing (from the user study).

5.1.1. General User Requirements for Physical Selection. The scenarios indicated need for each of the three selection methods. In an environment with many tags, it will be hard to select the correct one from the list presented after scanning. Therefore, there is a need for touching and pointing when the user knows the approximate location of the tag.

The analysis of the scenarios revealed that there should be a default action depending on the type of the tag read. If the tag contains for example a URL, the terminal should open the web browser and display the page. This means that a) there should be a tag-reading application running in the terminal all the time, and b) it should be able to determine an action for (most) tags the user selects. This also indicates that tags may include meta-information about their contents. Physical selection should make actions happen without the user first starting an application to interpret the tag contents.

5.1.2. User Requirements for Touching. There are three alternative (and possibly complementary) techniques for touching tags: a) touching is always on and available, b) touching is on and available after a “touch mode” is activated, and c) touching requires an additional specific action (for example a button press).

In the physical browsing user study, the users preferred touching being continuously on compared to touching with a button press. However, touching without a button could cause problems, for example if the terminal is put into a purse full of tagged objects as the terminal would then continuously select the tags.

The results indicate that the best technique for touching would probably be having a “touch mode” that can be activated and deactivated either manually or automatically. The alternative use of a button press confirming a touch action could be user configurable.

5.1.3 User Requirements for Pointing. Pointing is a directional selection method. There are three alternative and possibly complementary techniques for pointing at tags: a) pointing with a directed RF field, b) pointing with a visible pointing beam and c) pointing with an invisible pointing beam (for example infrared).

In the physical browsing user study, the pointing technique without a visible aid was more difficult than the techniques in which there was a visual aid.

The results indicate that a visible pointing aid is needed but there may still be a need for invisible pointing technology due to privacy concerns. To ease pointing, a visible pointing aid could be used with a wider invisible beam.

5.2 Gathering measurement data from wireless sensors

Gathering measurement data wirelessly was well illustrated in the scenarios. The scenario evaluations revealed user requirements for special cases such as long term data logging and using multiple devices. The user evaluations of the fitness and golf proof-of-concepts provided additional feedback on activating the measurements.

5.2.1 Physical selection in activating measurements. The initial vision in the scenarios was that every tag and sensor within the reach of the terminal device reading distance is continuously measured. Scenario evaluations with application field experts and user evaluations with the proof-of-concepts revealed that not all sensors can be read continuously as they only intermittently contain valid data. The user evaluations with proof-of-concepts indicated that physical selection can be utilized in activating measurements. E.g., the

user can get a single measurement or initiate a series of measurements by touching a sensor unit.

User feedback has several implications. Physical selection should activate proper application(s) to interpret and analyze the sensor reading. Each sensor reading must be addressed to an application that can validate and process the reading and control the sampling frequency. The validated measurement data can be made available for other applications as well.

5.2.2 Data logging in sensor units. In the scenario evaluations the interviewees told that during many sports activities, for instance when running they would prefer leaving the mobile terminal at home. They asked whether they could just carry the sensor unit and then later at home transfer the data to the mobile terminal.

User feedback implies that data logging in sensor radio node should be supported. This would require time stamping in sensor units. The architecture should facilitate data synchronization between data logging sensor radio nodes and the mobile terminal.

5.2.3 Long term monitoring. Most scenarios described only short-term monitoring. In the scenario evaluations application field experts pointed out the need for long-term monitoring (even years) for instance to get more insight to the symptoms of a patient.

These requirements suggest that the architecture needs to cooperate with external servers that can store the long-term monitored data.

5.2.4. Support for multiple user terminals. In the scenario evaluations the interviewees pointed out that one user may have several personal mobile terminals. For instance (s)he may use a wristop computer when jogging and a mobile phone when walking.

User feedback implies that the user should have seamless access to his/her exercise data with any device. The gathering of measurement data should continue fluently even if the user changes the terminal device temporarily or permanently.

5.3 Context-awareness

User feedback from the scenario evaluations emphasized user control and the role of context-awareness in activating situationally relevant applications. Field evaluations with the proof-of-concept that illustrated the use of context tags confirmed the potential of tags in activating contexts.

5.3.1. Context-based application management. User feedback from scenario evaluations and proof-of-

concept evaluations emphasized effortless use facilitated by context recognition. Initially we were focusing on context-awareness within individual applications but already when defining the scenarios we noticed that context-aware activation of applications was even more important. For instance in the golf scenario, the golf application activates as the user arrives at the golf course. Context-awareness within the application is presented in the same scenario when the golf application starts to record the swing as it recognizes the club near the wrist computer. Both kinds of context-awareness were repeated in our scenarios and widely accepted by the users.

User feedback implies that the architecture should take care of context-based application management, i.e. applications should be activated and deactivated according to the recognized contexts.

5.3.2. Tag-based context recognition. In the scenarios tag-based context recognition often turned out to be adequate because contexts are frequently related to certain places or certain everyday objects. Embedded tags can reveal the place or the objects around the user and thus the context. In the proof-of-concept evaluations the users accepted well context-recognition based on the user actively selecting a tag by physical selection because it saved user efforts.

User feedback implies that the architecture should support context tags: special tags that facilitate easy activation or even download of applications, or context-aware actions within an application. Contexts could be recognized more accurately if the architecture could measure distances and directions to the tags.

5.3.3. User control. In the proof-of-concept evaluations the users wanted to confirm most context-based actions. Only simple actions that are easy to cancel were accepted without user confirmation.

User feedback implies that the architecture needs to include elements that facilitate user control of context-aware features. Context validation may be required both in automatic and in user-initiated context recognition. Validation can take place by asking the user for confirmation or by reading additional sensors. Context-aware actions may require user confirmation as well as undo or cancel functions available.

5.4. Taking applications into use

Although our scenarios described ready-made installations, in the scenario evaluations the interviewees often referred to problems that they expected to face in installing and configuring the systems. Especially in relation to sports and fitness

scenarios, the interviewees would not have time for complex set-up operations. The interviewees also saw that if each single sensor would have to be activated separately, the user would easily forget some of them.

Taking an application into use requires installation of the application, introducing appropriate sensor units to it and configuring the setup. The usage patterns can be defined according to the temporal state of usage: a) the user connects the terminal and a sensor unit for one session, b) the user introduces the terminal and the sensor unit permanently for future sessions, c) the user activates the application, or the application activates itself automatically and d) the application connects itself to previously introduced sensor units.

5.4.1 Activating applications. In the scenario evaluations the users were worried about the installation of the applications that they expected to be complex. For many of the applications the interviewees could see only occasional usage, thus keeping the applications permanently active does not make sense. User feedback in the scenario evaluations confirmed that the user may use his/her personal mobile terminal for several different purposes. Each application may require a different set of sensors and tags, and the same application can be used with varying sets of sensors. Different applications can be active in parallel and by turns. The user may have applications that need to be continuously active, for instance health monitoring applications. The activation of other applications should not disturb these. The user evaluations with proof-of-concepts confirmed the potential of context tags as solutions to ease taking applications into use.

User feedback implies that the architecture should support the co-existence and fluent activation of situationally relevant applications. The architecture should support easy downloading and setting up of new applications as well as easy uninstallation of applications that the user no more needs. Installation and activation of applications can be made effortless to the user with tag-based context-awareness as described in the previous chapter.

5.4.2 Private sensors. In the scenario evaluations the users wondered how complicated it would be to connect the sensors wirelessly to the mobile device. They also were worried about how to make sure that nobody else is reading their personal sensors indicating for instance information on their health parameters.

User feedback implies that the architecture should support both public and private sensors. The architecture should provide the user with tools for the management of private sensor units. The tools should facilitate introducing the sensor units to the personal mobile terminal and the application on it, initiating a

measurement period and measuring. Introduction is needed to associate the sensors to the terminal and to a specific application, and to set up authentication and encryption parameters between the terminal and the sensor unit.

Using physical selection methods to introduce the sensors to the terminal device has good potential as it keeps to common usage patterns. Touching is a preferable method due to small range and minimal chance for eavesdropping. Scanning can be used in connection to setup wizards with user confirmations.

5.5. Ethical issues

The ethical issues raised up in the user evaluations were in line with earlier research [16]-[18]. We however found that many ethical issues can actually be dealt with at the level of the architecture, and the personal mobile terminal centric approach has potential to respond to the ethical challenges. The personal mobile terminal is a trusted device for personal data, providing facilities to ensure keeping the user in control. This constitutes a good basis for ethically acceptable solutions.

We shortly describe key ethical issues considering privacy, security, and trust that were arisen in the user evaluations and the scenario analysis.

1. What kind of information is retrieved of the user? The mobile terminal centric approach is focused on personal information (e.g. health measurements) that is very private to the user. Furthermore, context-related information such as nearby tags and personal context history are private information. Even if the user's identity cannot be directly revealed from the information, delivering it may be ethically arguable.

2. Who can get the information? The user should have the right to decide who can receive personal information. Physical selection by pointing and scanning, and generally reading from a distance, increase possibilities for eavesdropping. Privacy of the data communication should be protected by authorization mechanisms.

3. The user needs feedback when personal information is transferred. Tags are normally read to a certain application on the user device. However, there may be other tag readers around. The user should know if tags and sensors that (s)he wears or carries along are readable also to external readers and should get feedback if this kind of reading takes place.

4. The user needs effortless ways to protect valuable information. Personal data stored on the user's mobile terminal must be protected against theft and losing by providing back-up and authorization mechanisms. Connections to external servers need to be secure.

5. The mobile terminal and applications should be protected against external attacks. Context tags and downloadable applications may provide access to virus programs and other hostile attacks. The user needs means to ensure the reliability of downloadable applications and their providers.

6. Reliability limitations with a mobile terminal based solution should be considered. The terminal may get lost, the batteries may drain out, and server connections may break. Backup mechanisms are needed to handle these situations.

These issues highlight the user requirements of awareness, control, and feedback of data stored in or mediated via the personal mobile device. The requirements are important as personal data may be maliciously used to threaten the user's privacy or security. The user should not however be responsible alone for protecting against these threats, but ethical issues should be taken into consideration at the level of the architecture, in compliance with laws and common moral norms and rules.

6. Conclusions

Our experiences point out that several system level design decisions concerning a mobile terminal centric ubiquitous computing architecture have significant influences on the end user experience. Our experiences also point out that user requirements regarding the architecture can be identified early in the design by illustrating the forthcoming applications as scenarios and proof-of-concepts and by evaluating those illustrations with potential users. Rich scenario material is essential to identify common usage patterns and further use cases. User feedback needs to be analyzed thoroughly in cooperation with technical experts to interpret user requirements to implications on the architecture.

Our vision of mobile terminal centric approach to ubiquitous computing was well accepted among the users. The occasional usage needs identified in connection to many of the proposed solutions backed up our design approach of a low-cost infrastructure built on the user's personal mobile terminal and providing platform for several kinds of applications and services. However, user feedback changed quite a lot our initial vision of the functionality of the architecture.

With some functionalities, especially context-awareness, we were focused on overly complex solutions while user feedback proved that simple solutions were sufficient in most cases. Some functionalities turned out to be more complex than foreseen. Tag and sensor connection distance and

directionality need to be precise because proof-of-concept evaluations revealed that indefiniteness in them caused a confusing user experience. Once the connection has been set up, it should be kept on by tuning the reading distance.

The new user interaction paradigm of physical selection was well accepted as it enables the user easily to get information of the environment and near-by objects. Physical selection can also be used in user-initiated context recognition and in introducing personal sensor units to applications.

Ease of taking applications into use and ethical issues were repeated concerns of our interviewees and test users. Taking these issues into consideration at the architectural level helps to ensure that they are responded in all the applications and services built on the architecture. This will increase users' trust on the applications.

7. References

[1] E. Kaasinen, T. Tuomisto and P. Väikkynen, "Ambient functionality - use cases," in *Smart Objects Conference*, Grenoble, 2005.

[2] ISO 13407:1999, Human-centered design processes for interactive systems. International standard. International Standardization Organization, Geneva, Switzerland, 1999.

[3] MIMOSA project home page. www.mimosa-fp6.com

[4] S. D. Mainwaring, M. F. Chang and K. Anderson, "Infrastructures and Their Discontents: Implications for Ubicomp," in: N. Davies, E. Mynatt and I. Siiro (eds.), *UbiComp 2004: Ubiquitous computing. 6th International Conference. Proceedings*, 2004, pp. 418-432.

[5] N. Streitz and P. Nixon, "The Disappearing Computer," in *Communications of the ACM* 48(3), 2005, pp. 33-35.

[6] G. Cockton, "Value-Centred HCI," in *Proceedings of the third Nordic Conference on Human-Computer Interaction (NordiCHI)*, ACM Press, New York, 2004, pp.149-160.

[7] G. Gay and H. Hembrooke, *Activity-Centered Design: An Ecological Approach to Designing Smart Tools and Usable systems*. The MIT Press, Cambridge, MA, 2004.

[8] N. Islam and M. Fayad, "Toward ubiquitous acceptance of ubiquitous computing," in *Communications of the ACM*, 46(2), 2003, pp. 89-92.

[9] P. Ketola, *Integrating usability with concurrent engineering in mobile phone development*, Academic dissertation, University of Tampere, Department of Computer and Information Science, A2002-5, ISBN 951-44-5359-x, 2002.

[10] W. K. Edwards, V. Bellotti, A. K. Dey and M. W. Newman, "Stuck in the Middle: The Challenges of User-Centered Design and Evaluation for Infrastructure," in *Proceedings of CHI2003*. ACM Press, New York, 2003, pp. 297-304.

[11] H. Ailisto, L. Pohjanheimo, P. Väikkynen, E. Strömmer, T. Tuomisto and I. Korhonen, "Bridging the Physical and Virtual Worlds by Local Connectivity-Based Physical Selection," in *Personal and Ubiquitous Computing*, Online First, 2006.

[12] R. Ballaghas, J. Borchers, M. Rohs and J. Sheridan, "The Smart Phone: A Ubiquitous Input Device," in *Pervasive Computing* 5(1), 2006, pp. 70-77.

[13] 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*, ACM Press, New York, 1999, pp. 370-377.

[14] J. Riekkki, T. Salminen and I. Alakärppä, "Requesting Pervasive Services by Touching RFID Tags," in *Pervasive Computing* 5(1), 2006, pp. 40-46.

[15] Väikkynen, P., Niemelä, M. and Tuomisto, T. Evaluating touching and pointing with a mobile terminal for physical browsing. Accepted to Proc. NordiCHI 2006.

[16] M. Friedewald, E. Vildjiounaite and D. Wright, *The brave new world of ambient intelligence*. Deliverable D1. A report of the SWAMI consortium to the European Commission under contract 006507. June 2005. Available: <http://swami.jrc.es>

[17] S. Lahlou, M. Langheinrich, C. Röcker, "Privacy and trust issues with invisible computers," in *Communications of the ACM* 48(3), 2005, pp. 59-60.

[18] M. Ohkubo, K. Suzuki and S. Kinoshita, "RFID privacy issues and technical challenges," in *Communications of the ACM*, 48(9), 2005, pp. 66-71.