HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Computer Science and Engineering
Software Business and Engineering Institute

Antti Nummiaho

Using Context Information to Enhance Mobile Communications

Licentiate Thesis

In Espoo 26.11.2007

Supervisor: Prof. Marko Nieminen
Instructor: D.Sc. (Tech.) Timo Laakko

HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Computer Science and Engineering

ABSTRACT OF LICENTIATE THESIS

Author
Antti Nummiaho

Date
26.11.2007

Pages
80+8

Title of thesis
Using Context Information to Enhance Mobile Communications

Professorship
T-121 Usability Research

Professorship Code
ZD11

Supervisor
Prof. Marko Nieminen

Instructor
D.Sc. (Tech.) Timo Laakko

Context-awareness means knowledge of the user's physical and social surroundings, which enables the creation of richer applications. In my research I concentrate on different areas of context-awareness on the end user's point of view based on a literature study. I especially concentrate on context-aware communication. I discuss the possible uses of context information in practical context-aware communication applications and the way context information can be created by the users. I also discuss the means users have for obtaining context information of other parties and getting informed when their contexts change, as well as what privacy concerns arise from context-aware communication. Related research projects are presented and evaluated against the literature study. The findings of the literature study are also inspected against the 35 000 ideas for new mobile services that the idea movement of VTT Technical Research Centre of Finland collected in spring 2006. In addition to this, the ideas are also evaluated with users using a web-based questionnaire. Also, an application framework for mobile context-based messaging is developed. The questionnaire results indicate user preferences and provide many practical examples of utilizing the context-aware communication scenarios. Furthermore, the results are summarised to provide practical guidelines for the creation of context-aware communication applications. The application framework raises many questions on what should be taken into account when developing mobile context-aware communication applications.

Keywords: Context-aware computing, ubiquitous computing, mobile communications

TEKNILLINEN KORKEAKOULU Tietotekniikan osasto	LISENSIAATINTUTKIMUKSEN TIIVISTELMÄ

Tekijä	Päiväys
Antti Nummiaho	26.11.2007
	Sivumäärä: 80+8

Työn nimi
Kontekstitiedon käyttö mobiiliviestinnän edistämiseen

Professuuri	Professuurin koodi
T-121 Käytettävyystutkimus	ZD11

Työn valvoja
Prof. Marko Nieminen

Työn ohjaaja
TkT Timo Laakko

Kontekstitietoisuus tarkoittaa tietoa käyttäjän fyysisestä ja sosiaalisesta ympäristöstä, mikä mahdollistaa monipuolisempien sovellusten luomisen. Tutkimuksessani keskityn kontekstitietoisuuden eri osa-alueisiin loppukäyttäjän näkökulmasta kirjallisuustutkimukseen perustuen. Erityisesti keskityn kontekstitietoiseen kommunikaatioon. Käyn läpi kontekstitiedon mahdollisia käyttötapoja kontekstitietoisissa kommunikaatiosovelluksissa sekä miten kontekstitietoa voidaan luoda käyttäjien toimesta. Lisäksi käyn läpi mitä keinoja käyttäjillä on kontekstitiedon saamiseen muilta tahoilta ja miten käyttäjille voidaan kertoa näiden kontekstitietojen muuttumisesta sekä mitä yksityisyyttä koskevia huolia asiaan liittyy. Aihetta sivuavat tutkimusprojektit esitellään ja niiden tuloksia arvioidaan kirjallisuustutkimuksen pohjalta. Kirjallisuustutkimuksen havaintoja tarkastellaan myös VTT:n idealiikkeen keväällä 2006 keräämiä 35 000 uutta mobiilipalveluideaa hyväksi käyttäen sekä käyttäjille suunnatulla webbipohjaisella kyselyllä. Lisäksi kehitetään mobiilin kontekstitietoisen viestinnän sovelluskehys. Kyselyn tulokset osoittavat käyttäjien mieltymykset ja tarjoavat monia käytännön esimerkkejä siitä, miten kontekstitietoisen kommunikaation skenaarioita voidaan hyödyntää. Tuloksista myös edelleen johdetaan käytännöllisiä ohjenuoria kontekstitietoisten kommunikaatiosovellusten luomiseen. Sovelluskehys puolestaan nostaa esiin useita kysymyksiä siitä, mitä asioita tulee ottaa huomioon mobiileja kontekstitietoisia kommunikaatiosovelluksia kehitettäessä.

Avainsanat: Kontekstitietoinen tietojenkäsittely, mobiiliviestintä

Acknowledgements

This licentiate's thesis has been conducted as part of the research work of VTT's Mobile Interaction cluster's Mobile Context team.

I would like to thank my instructor D.Sc. (Tech.) Timo Laakko for his comments about the content of this study and on the structure of this thesis.

I would also like to thank thesis's supervisor Prof. Marko Nieminen for his guidance and support.

In addition to the above-mentioned, I would like to thank everyone who participated in the questionnaire, my friend Anni for drawing a nice picture to the introduction chapter, and anyone else that somehow affected on the creation of this thesis.

In Otaniemi 26.11.2007

Antti Nummiaho

List of Publications

The thesis is based on the following two publications:

I. **Nummiaho, A. (2007).** User Survey on Context-Aware Communication Scenarios. In Proceedings of the 1st International Symposium on Computer Human Interaction in Mobile Technology (IS-CHI 2007), Singapore, pp 486-489.

This paper evaluates thirteen different context-aware communication scenarios with 48 users using a questionnaire. The author was solely responsible for preparing the questionnaire and analyzing the results.

II. **Nummiaho, A. and Laakko, T. (2007).** A Framework for Mobile Context-Based Messaging Applications. In Proceedings of the 1st International Symposium on Computer Human Interaction in Mobile Technology (IS-CHI 2007), Singapore, pp 627-630.

This paper presents a framework for mobile context-based messaging applications. The author was largely responsible for designing and implementing the framework.

Abbreviations

API	Application Programming Interface
CPU	Central Processing Unit
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HTTP	Hypertext Transfer Protocol
J2EE	Java 2 Enterprise Edition
J2ME	Java 2 Micro Edition
JSR	Java Specification Request
LIF	Location Interoperability Forum
MMS	Multimedia Messaging Service
NFC	Near Field Communication
PDA	Personal Digital Assistant
RFID	Radio Frequency Identification
SMS	Short Message Service
Wi-Fi	Wireless Fidelity
WLAN	Wireless Local Area Network
XML	eXtensible Markup Language

Table of Contents

1. Introduction

This chapter begins with a short motivation of why the subject of this thesis is currently important and topical. After that, the existing research is briefly introduced and the research questions and methods of this thesis are discussed. Finally, the structure of this thesis is presented.

1.1 Motivation

According to The Free Dictionary (2007), context can be defined as "*the circumstances in which an event occurs; a setting*". In this work, the circumstances can be seen to consist of user's physical and social surroundings. Therefore, examples of context information include location, time, current activity, and identities of nearby people.

Communication is a process that allows us to exchange information by several methods. Especially nowadays, people communicate more and more using mobile devices. There were 2.3 billion mobile phone subscribers in the world in 2006 and the number is expected to increase to 3.3 billion by the year 2011 (Nikkei Electronics Asia, 2007).

Currently, mobile devices are not conveying very much context information to enhance the communication. However, the situation may change soon as the newest mobile phones are starting to support several ways of gathering context information. More and more GPS enabled mobile devices are appearing on the market including Nokia N95, Samsung SGH-i550, LG VX8700, and Motorola RAZR Maxx Ve. This will allow the tracking of location information. Market research predicts that 25 % of mobile phones in 2010 will be GPS-enabled (Reuters, 2007). Nowadays most mobile devices also support Bluetooth, which can be used to detect nearby people, who are also using Bluetooth-enabled devices. Also, NFC-enabled phones, like Nokia 6131, can use RFID tags to make it possible to detect contexts quickly.

Therefore, an interesting and topical question is how the addition of more context information would improve the communication with mobile devices. This thesis focuses on this question by discussing the possibilities of how context information can be used to enhance mobile communications. The issue will be examined both on the end user's and on the application developer's viewpoint.

1.2 The Potential of Context-Aware Systems

Researchers have come up with numerous context-aware systems that utilize location information. Examples of these include various tourist guides where information is displayed based on the current location. Location information can be gathered using a variety of sensors including GPS satellites, mobile phone towers, badge proximity detectors, cameras, magnetic card readers, barcode

readers, etc. These sensors can provide either position or proximity information. (Baldauf et al., 2007)

Systems that take advantage of also other context information than location have been developed as well. More adaptive and useful applications can be built using context elements such as noise, light, health, and mood. These applications are especially well-suited for specific conditions such as hospitals as they can be optimised for these specific environments without the need to make them flexible or extensible. (Baldauf et al., 2007)

Developing applications from scratch can be rather slow and require a lot of basic work that is common to all context-aware applications. Johnson (2007) has developed a generic framework to solve this problem. The framework provides a mechanism to quickly develop innovative context-aware applications by defining the applications with a mark-up script.

Context-aware communication applications have also been analysed. Schilit et al. (2002) provide a thorough discussion on the potential benefits of utilising context information in communication applications. These include determining which people should be included in a communication based on context, delivering messages in the most timely and relevant context, and using the shared context information to help the communication be polite but also productive.

Fogarty et al. (2005) have especially focused on studying how to use context information to quickly assess how interruptible a person is. This could potentially improve human computer interaction as poorly-timed, disruptive communication attempts would be avoided. An example scenario is presented in Figure 1.

Raento (2007) discusses privacy issues that arise from the sharing of context information. Most important aspect is that users must be able to control and see what context information about them has been sent to whom. Also, the value that the users get from revealing information must be clearly visible so that users can easily comprehend what they are gaining in exchange for exposing their context information.

Many prototypes, including Kontti (Kolari et al., 2004), ContextContacts (Oulasvirta et al., 2005), iCAMS (Nakanishi et al., 2004), InfoRadar (Rantanen et al., 2004), Live Contacts (Ter Hofte et al., 2004), Socialight (Melinger et al., 2004), and Context Watcher (Koolwaij et al., 2006) have been implemented and evaluated with users to examine certain features of context-aware communication.

This thesis will contribute to the existing research by providing a thorough examination of users' needs and concerns on mobile context-aware communication. It will also discuss some issues that the application developers should take into account when creating context-aware communication applications.

Figure 1. Using context information to avoid poorly-timed, disruptive communication attempts.

1.3 Research Questions and Methods

The goal of this research is to examine how context information can be used to enhance mobile communications. In the research I concentrate on the following research questions.

- How can the context information be used in a mobile communications scope?
- What advantages there are in using the context information in a mobile communications scope?
- What concerns there are in using the context information in a mobile communications scope?

The used research methods include a literature study on related research, inspection of 4000 out of the 35 000 ideas for new mobile services that the VTT Technical Research Centre of Finland collected in spring 2006, a web questionnaire that 48 persons answered, and the creation of an application framework for mobile context-based messaging applications.

These generic research questions will be refined after the related research has been introduced.

1.4 Structure of the Thesis

This thesis is divided into seven chapters. The contents of these chapters are shortly described and justified in the following.

The first chapter is this introduction, which began with a short discussion of why the subject of this thesis is currently important and topical. After these motivations, the existing research was briefly introduced and the contributions that this thesis will make were presented. Finally, the research questions and methods of this thesis were discussed.

The second chapter covers the backgrounds of context-aware computing. It begins with a short history to context-aware computing along with the definitions of context and context-awareness in order to familiarize the reader with the concepts relevant for this study. The chapter ends with the introduction of some typical context-aware applications and with the discussion of the issues related to developing context-aware applications. These issues are relevant as the thesis is mainly about analysing possible application ideas related to context-aware mobile communication. The issues also relate to the application framework for mobile context-based messaging applications which is being developed in this thesis.

The third chapter concentrates on related research on context-aware communication therefore providing the basis for the research. At first goals of context-aware communication applications are recognized. After that, ways for creating, representing, and obtaining context information are examined, and implemented context-aware communication applications and prototypes are introduced. The chapter ends with some concluding remarks of the related research. The issues presented in the chapter are used to formulate the questions for the user survey and to aid in the development of the application framework.

The fourth chapter narrows the focus of the thesis based on the related research and further specifies the research questions that this thesis focuses on. The different context-aware communication scenarios that the user survey focuses on are also extracted from the related research. The purpose of this chapter is to set up the constructive part of the thesis.

The fifth chapter along with Papers I and II presents the results of the research. Results of the analysis of the idea movement's ideas, questionnaire results as well as a presentation of the application framework are included. The chapter constitutes the main part of the constructive part of the thesis.

The sixth chapter includes an analysis of the results presented in the previous chapter. Purpose of the chapter is to find out how the results of the different research methods link together and what issues they raise regarding the scope of the research.

Finally, in the seventh chapter conclusions of what was presented are drawn, answers to the research questions are summarized, and future work is discussed.

The thesis ends by listing possible scientific forums for publishing the future results on this topic.

Appendixes of the thesis include the web questionnaire, questionnaire charts that summarise the questionnaire results, as well as an example application that uses the application framework developed in the thesis.

2. Background

This chapter introduces the backgrounds of context-aware computing. A short history of context-aware computing along with the definitions of context and context-awareness are given. Also, some typical context-aware applications are introduced and the issues related to developing context-aware applications are discussed.

2.1 History

Context-aware computing relates strongly to terms like ubiquitous computing, pervasive computing or ambient intelligence. The first term that described this area of computer science in its current sense was ubiquitous computing, which was first introduced by Mark Weiser in 1988 as he worked at the Computer Science Lab at Xerox PARC. The first paper that defined ubiquitous computing was the article “The computer for the 21st century” that was written for the Scientific American by Weiser (1991). In this paper Weiser presented a vision of a technology that would disappear to the background and, therefore, help users without distracting them. Mark Weiser wrote:

“Most important, ubiquitous computers will help overcome the problem of information overload. There is more information available at our fingertips during a walk in the woods than in any computer system, yet people find a walk among trees relaxing and computers frustrating. Machines that fit the human environment, instead of forcing humans to enter theirs, will make using a computer as refreshing as taking a walk in the woods.”

The term context-aware computing was then introduced by Schilit et al. (1994) as software that “adapts according to its location of use, the collection of nearby people, and objects, as well as changes to those objects over time”. Both terms pervasive computing and ambient intelligence were introduced in 1999. Pervasive computing was introduced by IBM (1999) as computing that “encompasses the dramatically expanding sphere of computers embedded within and intrinsically part of larger devices”. Ambient intelligence was introduced by Phillips as “digital environments in which the electronics are sensitive to people’s needs, personalized to their requirements, anticipatory of their behaviour, and responsive to their presence”. All these terms are nowadays commonly used, although ubiquitous computing (or ubicomp for short) is probably the most commonly used. Also, ambient intelligence is especially used in European Commission. Context-aware computing is seen as a key component of ubiquitous computing. Other less used terms include calm technology introduced by Weiser et al. (1995), things that think introduced by MIT Media Lab, transparent computing, physical computing, everyware, and tangible media. One might wonder why so many terms have been created to define basically the same idea. There is probably no single reason, but it seems that the main motive for creating a new term has usually been that people have felt that the earlier

terms have become too strongly associated with a certain viewpoint, institution, funding source, or dominant personality.

One of the first papers on context-aware computing research was written by Want et al. (1992). It discussed the Olivetti Active Badge - a system for the location of people in an office environment. The paper examined different location techniques, privacy concerns, and applications especially related to telephone call routing. One of the first examinations on how to build context-aware services was the PhD thesis of Bill N. Schilit (Context-Aware System Architecture for Mobile Distributed Computing) (Schilit, 1995).

2.2 Defining Context

There have been many attempts to define what context is. Some have defined context by categorizing and listing possible context elements. Others have stated that it is impossible to list all possible context elements and, therefore, context must be defined in a more abstract way. Clearly, both ways have their uses. Listing context elements provides insight into what context consists of and abstract definitions can be used to check whether a certain information is context information or not.

Let's first take a look on two different attempts to categorize and list different context elements.

Schmidt et al. (1999b) provide a hierarchical listing of context elements:

- Human factors
    - User related (habits, emotional state, biophysiological conditions etc.)
    - Social environment (co-location of others, social interaction etc.)
    - Tasks (spontaneous activity, engaged tasks, general goals etc.)
- Physical environment
    - Location (absolute position, relative position, co-location etc.)
    - Infrastructure (computational resources, communication etc.)
    - Physical conditions (noise, light, pressure etc.)

A slightly different way to divide and list context elements is provided by Göker et al. (2002). They take into account the importance of considering whether certain context information is of static or dynamic nature:

- Task related factors (goals, subtasks, actions, activities, events etc.)
- Social factors (friends, enemies, neighbours, co-workers, user's role etc.)
- Personal factors
    - Physiological factors (pulse, blood pressure, weight, hair color etc.)
    - Mental factors (mood, expertise, angriness, stress etc.)
- Spatio-temporal factors (time, location, direction, speed, place, clothes etc.)
- Environmental factors (services, temperature, light, noise, persons etc.)

Although these categorizations may seem quite different, they are similar in a way that both identify personal attributes (physical and mental), social aspects, task related factors, and physical surroundings as the major components of context. Also, as we can see from the listings, there are countless elements that can be seen as context information. Therefore, creating a commonly understood categorization of context elements may prove to be very difficult if not impossible. Some elements may even have cultural or geographical dependency and even if some elements like lux and decibels can be measured, they are hardly ever discussed in exact terms in everyday life, which would make them very difficult for the users to specify. (Hiltunen et al., 2005)

Let's now take a different approach and present a formal definition. The most notable formal definition of context comes from Dey et al. (2000). They write:

"Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves."

As this thesis concentrates more on context-aware communication's practical scenarios rather than theoretical issues, the approach of listing the used context elements is mostly utilized from here on whenever there is a need to discuss what is meant by context.

2.3 Defining Context-Awareness

Dey et al. (2000) provide a thorough discussion on defining context-awareness. They also argue that previous definitions of context-awareness have been too specific and provide their own definition:

"A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task."

In general, context-awareness can be seen as a process of three phases:

1. Deduce the context
2. Process the context
3. Use the context

Ideally, context can be deduced without bothering the user. The primary way of achieving this is the usage of different sensors. There are many types of sensors that can be used including (Mayrhofer, 2004):

- Vision (e.g., cameras and video cameras)
- Audio (e.g., microphones)
- Location (e.g., GPS, GSM cells, WLAN, Bluetooth, and RFID)
- Orientation (e.g., gyroscopes, magnetic field, and tilt sensors)
- Proximity (e.g., Bluetooth, WLAN, RFID, and touch sensors)

- Environmental conditions (e.g., temperature, humidity, and air pressure meters)
- Identity (e.g., iris scanning, fingerprint sensors, RFID, and infrared badges)

The main difficulty in deducing the context is that the context is constantly changing and the change is often gradual and hard to predict. Also, not all information can be easily obtained through sensors. For example, getting information about user's feelings could be difficult to accomplish automatically. In some cases it may be useful to store context history and use it to predict the future. Also, shared context can be used to take advantage of what others have done in similar situation. (Mayrhofer, 2004)

The context is processed to extract relevant information and that information is then used to change the application's behaviour in some meaningful way. Chen et al. (2000) suggest that the usage of context can be divided into active context-awareness and passive context-awareness. An actively context-aware application automatically adapts to the new context by changing the application's behaviour and a passively context-aware application presents the new context to the user and leaves the decision on whether the application's behaviour should change explicitly to the user.

2.4 Context-Aware Applications

Context-aware applications have been developed since 1990's. At first, location was the only source of context information, but soon researchers started to investigate other possibilities as well.

2.4.1 Early Applications

The first well-known context-aware application was Active Badge developed at Olivetti Research Ltd. It used user's location in a building as context information. Location was presented to the receptionist, who then forwarded incoming telephone calls to the user's nearest phone. The system was later updated to forward calls automatically. It was also found out that users preferred having control over when calls were forwarded to them so that they wouldn't be interrupted for example in the middle of a meeting. (Want et al., 1992)

Based on the Active Badge locating system a Teleporting application that used user's location and the location of workstations to dynamically map the user interface to whatever computer that was near the user at any time was also developed at Olivetti Research Ltd. It enabled the application to follow the user when he moved around. (Frazer et al., 1994)

Another location based application was Active Map developed at Xerox PARC. It tracked people's positions on a map at all times, which enabled the quick locating of people in an office building. (Want et al., 1995)

Shopping Assistant was created at AT&T Bell Laboratories to use customers' location in a store to guide the shoppers. The system provided details of items, helped to locate items, pointed out items on sale, compared prices, and so forth. (Asthana et al., 1994)

Cyberguide developed at Georgia Institute of Technology was another classical example of context-aware applications. It used tourists' location and time to provide information services about their current location for them. It enabled for example suggestions for directions, background information on specific places, and the ability to leave comments on an interactive map. It also gathered history information of the visited places to a travel diary and used it to suggest places of interest to visit. The location information was collected outdoors by GPS and indoors by an infrared positioning system. (Abowd et al., 1997)

Several other tourist guide systems were also developed including the GUIDE system developed at University of Lancaster for the visitors of the city of Lancaster (Davies et al., 1999). The GUIDE system was also later extended to better support co-operation between city visitors by enabling them to share their experiences and associated context information (Cheverst et al., 2000). Also, many similar systems were developed for example for museum visitors and exhibition tourists (Chen et al., 2000).

Brown et al. (1997) in university of Kent at Canterbury implemented a People and Object Pager which used user's location and information of nearby people and objects to for example broadcast a request to locate a certain book and whoever encountered the book was notified to pick it up for the requester.

Conference Assistant developed at the Georgia Institute of Technology used user's location, current time, presentation schedule, and topics as well as user's interests to suggest presentations for the user to attend. The system also automatically displayed the name of the presenter, title and slides of the presentation, and other related information when a user walked into a presentation room. (Dey et al., 1999)

Adaptive GSM phone and PDA developed at TEA (Technology for Enabling Awareness) at Starlab used user's activity, light level, pressure, and proximity of other people to (1) adapt the font size to the light level and user's activity (a large font when the user is walking, small font when stationary) and to (2) automatically set the appropriate profile (ring, vibrate, adjust the ring volume, or keep silent) for the mobile phone depending whether the phone was in hand, on a table, in a suitcase, or outside. (Schmidt et al., 1999a)

Office Assistant developed at MIT Media Laboratory took advantage of user's current activity, schedule, and the detection of an approaching visitor to adapt its behaviour based on the identity of the visitor. (Yan et al., 2000)

ComMotion was also implemented at MIT Media Laboratory to use both location and time to drop reminder messages on locations. When the intended recipient arrived at the location, the message was delivered via voice synthesis. (Marmasse, 1999)

CybreMinder developed at Georgia Tech was a similar system that used more context information such as nearby people and current weather conditions. (Dey et al., 2000)

A more thorough discussion of the first context-aware applications can be found in the paper “A Survey of Context-Aware Mobile Computing Research” (Chen et al., 2000).

It is interesting to notice how many of the ideas present in these early applications are still relevant research topics nowadays. For example, Cyberguide and GUIDE enabled users to leave messages on different locations, which is also the main functionality of the application framework developed in this thesis. In addition, ComMotion and CybreMinder studied context-based reminders, which are also examined in the web questionnaire of this study.

2.4.2 More Recent Applications

More recent context-aware applications have focused more on certain environments where the use of context information could potentially have many useful benefits. Two very common environments include homes and hospitals.

For example, Meyer et al. (2003) suggest the following context-aware home scenarios:

- Lights, chairs, and tables automatically adjust as soon as the family gathers in the living room to watch TV.
- Phones only ring in rooms where the addressee is actually present, preventing other people being disturbed by useless ringing.
- The music being played in a room adapts automatically to the people within and the pictures in the frames on the desk change depending on which person is working there.
- In-house context-aware communication systems allow family members to speak to each other as if they were in the same room, even when they are in different rooms.
- Complete security systems including emergency call out alarms for burglars, fire, or injury with a complete awareness of the home owners wherever they are.
- In assisted living complexes, context-aware systems monitor the state of the elderly occupants, freeing the nursing staff from the task of constantly supervising them.

For hospital environment it has been suggested that location could be used to reduce the information overload so that only information that is useful and relevant for a certain location is delivered. For example, a nurse doesn’t need to know the appropriate dose of medication until she must give it to the patient. Likewise, the nurse might want to locate the closest cardiologist to perform an emergency check-up. Time information could be used to deliver messages in appropriate times. For example, a doctor might leave a message that describes

recommendations for treatment to any nurse on the next shift since then the patient's symptoms have had sufficient time to evolve. If the messages are not sent instantly, the users are also enabled to modify or delete the messages they have "sent", if conditions change. Also, in hospitals work is more based on roles than particular individuals. Therefore, it should be possible to address messages to certain roles like to "the nurse on the afternoon shift," or "the next doctor to visit the patient." (Munoz et al., 2003; Bardram et al., 2004)
Medical applications could also utilize reminders in reminding the patients to take their prescribed medication. Context information could be used to deliver the reminders in the most optimal situation in contrast to the reminder systems used nowadays where the reminders are delivered only based on time. Instead of interrupting the user's activities, the reminders could be subliminal audio and visual cues that lie below the user's threshold of perception but are just enough to jog the user's memory. (Pentland, 2004)

Miller et al. (2004) have also studied how to deliver medical reminders to the user in a polite and comfortable way. They conclude that the level of politeness should adapt based on each individual's expectations and actions. On the other hand, if the reminder is considered helpful in a certain situation, even a certain level of impoliteness might be tolerable.

Different rules can also be used to determine the right action based on context. For example, if a person suffers an epilepsy seizure, the system could direct the nearest caregiver to his/her location and based on the severity of the seizure determine whether only voluntary or also professional caregivers should be contacted. (van Sinderen et al., 2006)

Baldauf et al. (2007) also discuss some of these more recent context-aware applications in their paper "A Survey on Context-Aware systems".

For the purpose of this thesis, it may be useful to try to generalize these focused application ideas to cover context-aware communication in a wider scope. For example, medical reminders are just one case of context-aware reminders and the idea of trying to disturb the user as little as possible also applies to context-aware reminders in general.

2.4.3 User Needs

Brown et al. (2002) note that the user is likely to be more interested in the context "just ahead" rather than the current context. They call this the context-of-interest. For example, a tourist might be interested in retrieving information just before he needs it.

One way to predict upcoming contexts is through collecting context history. Wilson et al. (2005) have analysed the use of context history in the domain of automatic health monitoring and found it very useful in determining the current activity.

Brown et al. (2002) also emphasize that users of mobile devices are likely to be occupied in other activities and, therefore, applications should not bring irrelevant information to users' attention and even relevant information should be presented to the user in a manner that does not interfere too much with their other activities.

Generally, adapting the user interface based on the user's context could make the application more efficient to use. For example, when an application notices that a person is engaged to an instant messaging activity, it could offer quick ways to initiate other communication methods, like phone calls, as well since one use for instant messaging is to establish the person's conversational availability. This could enable the person to initiate a phone call quickly without the process of dialling a number, waiting for the ring, and having the other party to pick up. (Tang, 2007)

Another major concern for users is privacy. Although people often share their context when they are called (e.g., "I am in a meeting, could you call me later?"), they want to be in control of what is visible for others about them. People also want to know what others know about them and they like to share information selectively. Privacy concerns must be taken into account from the very start of the application development since they have killed many potential applications. (Schmidt et al., 2000)

2.4.4 Application Development

To achieve efficient application development, it should be possible to build new applications using existing components. The distributed nature and the use of unconventional sensors make this hard but even more important in context-aware applications. Already in 1999 this matter was noticed by the development of the Context Toolkit (Salber et al., 1999). It introduced context widgets for sensing presence, identity, and activity of people and things. These context widgets could be used as building blocks for new context-aware applications and they also helped to separate the handling of context from the actual functionality of the application. Since the toolkit only supported a small number of context elements, it could not support a large variety of different applications.

Korpipää et al. (2003b) have also created a framework for gathering context information systematically from the use's surroundings, processing it, and delivering it to the applications. They use systematic context ontology to define contexts that the applications can use while also taking into account that some contexts may be application-specific. The framework promotes the reuse of contexts and modules that produce them. The biggest problems with the framework relate to the huge amount of context information needed to reliably discriminate between different high-level contexts. This could consume the resources of the mobile device very quickly. Therefore, it is important to create the optimal strategies for gathering the context information.

A more recent framework for developing mobile context-aware applications has been presented by Johnson (2007). It introduces the concept of a context

fingerprint which is a characterisation of the context that a mobile terminal can determine from the sensors available to it. The framework supports a large number of sensors including GSM, Wi-Fi, Bluetooth and GPS radios as well as cameras, RFID and motion sensors. It also uses operating system APIs to get information like battery level, memory and CPU usage, applications running, and call details. Multiple context fingerprints can be defined, together with events and actions that are fired on transition between contexts. All that is needed for the creation of a context-aware application is defining the key elements in a mark-up script and loading the script to an implementation of the framework. However, as with the framework created by Korpipää et al. (2003b) a challenging issue raises from the fact that context-aware applications tend to produce lots of sensor data. For example, the more accurately location is tracked, the more data is generated. The framework expects a lot of resources from the mobile device since all the sensor calculations are done on it. Another approach would be to transfer all or some of the sensor data to a server and do the calculations there, but that approach also has its issues like how expensive would it be to transfer the data and would the users be willing to allow a huge number of private sensor data to be submitted to the network.

In general, the key principles for developing context-aware applications are to separate the context handling from the application's main functionality, to make the application extendable allowing new context elements to be easily added, and to create prototypes to be tested in real-life usage situations. Separating the context handling from the application's functionality eases the burdens of programmers and small devices and supports the creation of different applications. Making the application extendable is important, because devices and sensors might come and go, and user requirements typically change. Real-life prototyping enables the testing of the context-awareness property of the applications. (Law et al., 2006)

Users are also likely to invent ways of using the application that the designers had not thought of at all. This is because typically the conceptual models of the designers do not completely match the mental models of the users. For example, users started to use a system designed for asynchronous location-based messaging also for synchronous instant messaging. This happened because users found more need for that kind of chatting functionality and once some users started to use the application that way others followed. Problem with this kind of behaviour is that the application most likely is not the most optimal solution for purposes that it has not been designed for. On the other hand, it may reveal potential ways for further development of the application that the designers would not have otherwise thought of. These kinds of differences between what designers had thought of and how users actually use the application can only be caught by testing in the real-life. Therefore, it is very important to arrange usability tests with a small group of actual users before making the application available for all users. (Burrell et al., 2002)

The application is also usually needed to operate on multiple platforms that have different characteristics. Desktops, for example, are typically used continuously while the use of mobile phones is more task-focused and hurried. This implies that the application must be designed separately for different platforms from the

beginning and not only converted from a finished product. However, if the same application is to be used on different platforms, some amount of consistency is also required to make it easier for the user to switch between the platforms. (Tang et al., 2001)

Sometimes applications are not developed from scratch, but existing applications, like telephony, are being changed to support context-awareness. In the transitional phase the application should offer similar ways to co-operate with users that already have context-aware capabilities and with users that do not. This is because users may not be willing to use several user interfaces for similar actions. (Milewski, 2000)

All in all, context information can be utilized to make the application more efficient to use by enabling better adaptation to users' needs. On the other hand, the application must take privacy issues into account to have even the possibility of becoming a success. These issues are examined in the web questionnaire part of this thesis. For making the development of context-aware applications more efficient the idea of using a general framework has proven useful. It is also utilized in this thesis as a framework for mobile context-based messaging applications is developed.

3. Related Research

This chapter introduces the related research on the topic of context-aware communication, which can be defined as "the class of applications that apply knowledge of people's context to reduce communication barriers" (Schilit et al., 2002). At first, goals of context-aware communication applications are recognized and discussed. After that, ways for creating, representing, and obtaining context information are examined. Finally, examples of implemented context-aware communication applications and prototypes are introduced.

3.1 Goals of Context-Aware Communication Applications

The following goals can be recognized for different context-aware communication applications. (Schilit et al., 2002; Ranganathan et al., 2002)

- Right message at the right time
- Reminders only when they are actually relevant
- Sharing awareness of one's context
- Reducing ambiguity

Each of these is now examined in more detail.

3.1.1 Right Message at the Right Time

Right message at the right time means determining which people should be included in a communication based on context. It is about deducing the contextually appropriate people. This deduction process may take into account the contexts of both sides of the communication. It may also concern multiple parties, e.g., a mailing list that consists of personnel currently inside a certain building. (Schilit et al., 2002)

Also, context information can be helpful in determining the appropriate device and the preferred communication format for the communication at a certain point of time. Typically, users possess multiple communication devices, but may have access to only a subset of them at a particular time. Also, each communication device has its advantages and disadvantages depending on the user's preferences and the situation that the user is in. For example, a call to a cell phone might be appropriate when the user is on the move, but an instant message or SMS more appropriate when the user is in a theatre or in a meeting. (Ranganathan et al., 2003; Lei et al., 2004)

Since there are so many communication methods nowadays that allow us to communicate everywhere and anytime, it is often expected that the contacter should always be able to reach the contactee. However, there most certainly are times when the contactee does not want to be disturbed. Therefore, the ability to

intelligently and efficiently restrict or even filter out communication attempts based on context information becomes important. (Goertz et al., 2004)

Lei et al. (2002) have built a Notification Dispatcher system for the purpose of routing messages to several possible communication devices based on the recipients' preferences, context, and urgency of the message. They use instant messaging online status and calendar events as context information to base the routing decision on. An important aspect of the system is that the urgency of the message is taken into account so that high-priority messages are delivered instantly regardless of the context.

MyConnector (Danninger et al., 2006) studied the use of different context cues for predicting the availability of the receiver so that the sender could make contact at the right time, in the right context, and with the optimal communication medium. Its results indicated that learning a person's availability is a very hard task, because it is a highly personal characteristic and many activities occur spontaneously. However, time of day and location might indicate availability rather well at least for people with regular daily schedules. In addition to that, urgency and importance of the current task seem to be important availability cues. Users also tend to be more interruptible towards the end of an activity. Still, Danninger et al., (2006) conclude that it might be best to use some semi-automatic way to deduce the availability, i.e., have the users manually confirm their availabilities.

The use of multimodal context information for predicting users' availabilities has also been studied (Malkin et al., 2006; Fogarty et al., 2005). Especially image and audio elements are a natural choice as most mobile devices today have cameras and microphones. Using multimodal context information would make it possible to predict users' availabilities without bothering the users at all. This would free the users from having to input calendar information or change their online status constantly. Malkin et al. (2006) studied the use of periodic still images, but did not found them very helpful in predicting the availability. However, they concluded that this may have been due to the small number of data. On the other hand, they found out that the use of acoustic information proved to be a quite good indicator of the user's interruptibility. This is also noted by Fogarty et al. (2005), who suggest that recognizing whether someone is speaking in the room is a quite good indicator of how interruptible a person is. The silence detectors they used adapted quite well to background noise in office environments, but they were unclear whether the detectors would work in noisier environments.

It must also be noted, that it increases the contacter's burden if s/he has to actively inspect the contactee's availability information. Therefore, the benefits of doing this must be clearly visible for the contacter. For example, if the contacter and the contactee work in the same place, displaying the contactee's workload in an easily processable format may be a good enough incentive for the contacter to be polite in his/her communication attempts. On the other hand, if the contacter is an outsider, s/he may not consider the workload information that important. Therefore, the displayed availability information should always be

chosen so that it is relevant to each and every contacter individually. (Dabbish et al., 2004)

Another approach is to allow people to specify what they would like to discuss and with whom and let the application find the most appropriate communication method and moment based on everybody's context. This requires that the application maintains an awareness of its users' activities, preoccupations and social relationships at all times. It could be extremely complex to track all this, but it can be helped by concentrating only on a certain environment, e.g., meeting situations in office environments or living rooms at home. A single room can be quite easily equipped with various sensors including cameras and microphones to enable the gathering of context information in there while wearable sensors can be used to gather mobile users' context information. The application must also have access to various communication devices and methods to be able to initiate the communication in the best possible way. The different communication devices could include mobile phones and desktop computers or laptops but also, e.g., visual displays projected onto some convenient surface or directed audio. Allowing users to specify what they would like to discuss could be used to, e.g., block all incoming communication attempts unless they are related to the desired topic. Also, users could define that they would like to discuss some subject with certain people before a certain deadline and the application would find the appropriate time for the users. (Danninger et al., 2005)

The "digital secretary" type functionality described above would intelligently connect people based on the contacters' current situations and motivations to initiate the communication and the contactees' current situations and availabilities. This would improve the typical context-aware communication scenario where the context information of the contactee is conveyed to the contacter and the decision of whether to initiatiate the communication is left to the contacter. For the contacter, the parameters that need to be considered are the following.

- Message type (question, answer or information and whether it is a reply or a new message)
- Message urgency (e.g., a message concerning an upcoming meeting is likely to be urgent)
- Message complexity (how much effort is it likely to take from the contactee)
- Message matter (if it concerns the current context of the contactee, the message could be delivered instantly)
- Relationship between the contacter and the contactee (messages between friends or co-workers could be given more importance than messages from strangers)

For the contactee the corresponding parameters are:

- Interruptibility (whether in a meeting or already engaged to a conversation either by phone or face-to-face)

- Workload (if, e.g., doing something that requires considerable attention like programming)
- Current task (if the message relates to the current task, it could be delivered instantly)
- Relationship between the contactee and the contacter

The application must compare and match the parameters of both sides to improve the efficiency of the communication. Some parameters like the urgency of the message can be determined independently of the communication partner while others like message matter and contactee's current task need to be compared with each other to determine what effect they may have. The application must then inform the contacter whether the message was delivered instantly or delayed. The best moment to deliver delayed messages is usually between tasks, so the application must constantly track the changes in the contactees' tasks. (Gross et al., 2006)

The main benefits for delivering messages based on contexts are that the sender can write and send messages immediately when the idea of writing the messages arises and that the recipient can react to the messages immediately, because they are delivered to the right context. This corresponds to the current behaviour of saving messages as drafts until a suitable moment comes to send them and having to remember to check received messages constantly for possible actions they might need. Also, knowing the delivery context might help the sender to create more meaningful messages. (Jung et al., 2005)

The need to check and possibly modify or remove outgoing messages also becomes important when the messages are not delivered instantly but based on contexts. Confirmations of delivered messages become more useful for the sender, but could be manually sent by the receivers in order to protect their privacy. (Jung et al., 2005)

All in all, using context information to deliver messages at the right times provides value both for the contacter and the contactee. The contacter does not have to wait for the appropriate moment to initiate contact and the contactee avoids inappropriate interruptions.

3.1.2 Reminders Only when They Are Actually Relevant

Reminders are messages that are sent to inform someone about some future activity that s/he should engage in (Dey et al., 2000). Sending reminders only when they are actually relevant is an interesting concept. For example, who hasn't forgotten the grocery list on the refrigerator door when having gone to a grocery shop? Wouldn't it be nice if the list would be delivered when one arrived at the shop? Most people use calendars, post-it notes, e-mail, etc. to store their reminders and to-do lists, but this brings out the need to regularly check these places. It would make life easier, if information were delivered in the most timely and relevant context. Most research on this topic has gone on location based reminders. For example ComMotion (Marmasse, 1999) and CybreMinder (Dey et al., 2000) allowed users to associate to-do items with locations in the real

world. When the user was in the specified location, an audible cue was played and the user could inspect the relevant text or audio item. (Schilit et al., 2002)

PlaceMail (Ludford et al., 2006) studied location based reminders and lists to support everyday tasks. One of its findings was that the typical round circle was not sufficient for a location, but to efficiently place the messages to locations the users' movements and the locations' geographical layouts had to be known. The desired delivery point depended also on factors such as users' plans, which routes they took and whether there were lots of social distractions on the route. It was also found out that location based lists like shopping lists should be manageable, i.e., it should be possible to add or remove things from the list and the list should be simultaneously accessible by multiple users. The system should also be able to detect if the user is already planning to go to a place in which case a reminder is unnecessary. This would require the system to properly analyze the message contents.

Zhou et al. (2005) also studied how people's concepts of places correspond to actual physical locations. They also found out that single points with radii are not very good matches for people's understandings of places. Instead they suggest that the following shapes are the most common.

- Multiple dots (e.g., any McDonalds)
- Region (e.g., neighbourhood)
- Path (e.g., favourite walk)

Also, in general the application should support personal representations of locations since different people view places in different ways. The application should also provide interactive techniques and social collaboration for acquiring the locations. A well-designed interactive user interface can make it easier for the users to define the locations. Since collaborating users typically have some shared places, an already defined place could be used at least as a starting point for defining a new place. However, as each user may understand the place a bit differently, the application should allow these place definitions to be changed. (Zhou et al., 2005)

3.1.3 Sharing Awareness of One's Context

Sharing awareness of one's context is a need that arises from the things that the person who is making the decision whether or not to approach a possible communication partner takes into consideration. These include the following. (Schmidt et al., 2000; Nagel et al., 2001; Nakanishi et al., 2000; Tamminen et al., 2004)

- How important is it for me to communicate now?
- Where is the communication partner located?
- What kinds of resources are present?
- What is the communication partner doing?
- Who is the communication partner with?

- How convenient does it seem for the communication partner to be interrupted?
- What is the relation between the communication partners?
- What type of conversation will it be (important to whom, how long will it take, etc.)?
- How likely the communication partner is to know an answer to a certain problem?
- Is it socially acceptable to start a conversation on a certain topic in this situation?

Also, the recipient has to somewhat consider these things before, e.g., answering a phone call. Especially, the importance of the caller's message typically plays a significant role in the recipient's choice of whether to answer the call or not. (Schmidt et al., 2001)

De Guzman et al. (2007) have studied what context information the callers and receivers actually consider or wish to be considered when making or answering a phone call. Results show that the callers consider receivers' tasks and physical activities more than their locations or social availabilities. On the other hand, receivers would like the callers to consider all of their context information more and in particular their social availabilities more often and their locations less often.

Usually, all of these issues cannot be found out before the conversation is initiated. This leads to the inefficient questioning phase in the beginning of the conversation where the missing context information that is considered relevant is queried. For example, questions like where are you, what are you doing or are you able to speak, are very typical in the beginning of a phone conversation. (Schmidt et al., 2000)

Context information could also be used to set appropriate parameters for the conversation like initial volumes based on the knowledge of ambient sound levels in the locations of the caller and the recipient. Also, one's context may change during the conversation, which would also be convenient to find out by the other party. For example, if someone enters the room where one is having a phone conversation, one might become more reluctant to discuss private matters. (Nagel et al., 2001)

The acquired context information helps the communication to be polite but also productive. For example, it is unlikely one will get an answer to a personal question during a business meeting, or an answer to a business question during a family dinner. (Schilit et al., 2002)

Therefore, the users should be enabled to restrict the access to the context information based on different characteristics, e.g., who is asking, for what purpose, and what is the current context (especially time and location). However, this should be accomplished with as little interaction with the users as possible. It would be too intrusive to ask the users what should be done every time someone tries to access their context information. Myles et al. (2003) have designed a

system to address these issues through different privacy policies. (Myles et al., 2003)

Nakanishi et al. (2000) have created a system that shares the communication context (location and schedule). Their results indicate that knowledge of the communication context decreases the number of unanswered calls and, therefore, provides smoother communication for the users. On the other hand, knowledge of the communication context did not motivate to write email messages instead of calling. Therefore, they conclude that sharing the context information does promote changes in the timing of communication but not with the medium of the communication.

Sharing an awareness of one's context can also create serendipitous opportunities for communication. For instance, seeing a "buddy" go online on Microsoft Network (MSN) Messenger may tempt one to initiate communication spontaneously (Schilit et al., 2002). Another example would be that broadcasting one's context to other people in a geographic area might discover people not previously known to be close-by (e.g., "is anyone going for a lunch?") (Paciga et al., 2005).

In chat environment sharing an awareness of one's context helps seeing what the other person is doing and what is happening in his immediate surroundings. For example, it may help seeing whether the other person is paying full attention to the conversation. This may help decide whether to continue the conversation or chat again later if the other person seems too busy. For example, ConChat tried to find out how busy the user was based on what other applications he was running. However, it may be difficult to accurately determine how busy the user really is. (Ranganathan et al., 2002)

Fogarty et al. (2004) evaluated a context-aware communication client with 26 users in a 4 week study. Their most important finding was that sharing awareness of one's context could in fact be interpreted as a sign of presence instead of a sign of availability and, therefore, it is questionable whether it would reduce inappropriate interruptions. It was especially found out that knowing that a colleague was present, but not available, did not discourage users from sending instant messages. One possible reason could have been that the users considered instant messages as a non-intrusive way of communication, although as a whole they constituted a significant intrusion. The recipients also probably thought it impolite to complain about a single instant message sent at an inappropriate moment. One possible solution could have been to design the user interface so that the unavailability of the users was highlighted more over the fact that the users were present in the building.

Altogether, sharing awareness of one's context can be helpful in determining the person's availability. It can also make the communication more efficient. Knowing other persons' context information can also provide topics for the communication while, on the other hand, help avoiding inappropriate issues.

3.1.4 Reducing Ambiguity

Reducing ambiguity arises from the need to ensure that all parties mean the same thing when they say something. Semantic conflicts are typically caused by different contexts. For example, if one writes an e-mail on Tuesday saying let's meet today, but the recipient does not read the e-mail until the next day, he may mistakenly think that the meeting day is Wednesday, if he doesn't realize to check the send date.

There are basically three different causes for the ambiguity (Ranganathan et al., 2002):

- Naming conflicts occur when naming schemes of information differ significantly (for example, when synonyms are used to refer to the same thing).
- Confounding conflicts occur when information items seem to have the same meaning but in reality they have different interpretations (for example, a "hot" dish can mean that the dish is spicy or that its temperature is high).
- Scaling conflicts occur when different reference systems are used to measure a value (for example, price being measured in different currencies or time being different in different time zones).

Naming conflicts and confounding conflicts tend to require exhaustive knowledge of the context and the available terms to be resolved by software, although some of them, like the format for a date being different in USA (mm/dd/yy) and Europe (dd/mm/yy), can be identified rather simply. Scaling conflicts are usually slightly easier, if the sender's and recipient's location and the units used in those locations are known. Generally, the best way for the application is still to flag potentially ambiguous terms and possibly suggest what they would mean in the recipient's context. However, totally replacing these terms might be dangerous, since the sender may have already considered recipient's context when he has constructed the message. (Ranganathan et al., 2002)

Mankoff et al. (2005) have created an architecture that supports the building of context-aware services where the contexts may be ambiguous. They conclude that the applications should provide several redundant techniques to ensure that the contexts are unambiguous. This is because users should have the possibility to choose the appropriate technique based on the seriousness of the errors and the user's level of engagement in the task. Also, if one technique fails, it is crucial to have other alternatives. Another important conclusion is the importance of defaults especially in highly ambiguous contexts, where it would be too big of a burden for the users to ask everything from them. In addition, the application should ask the user only when necessary. If the application does not need to act on the data or it can act with possible ambiguities, it should do so and bother the user only when absolutely needed.

3.2 Creating Context Information

There are basically two options for the end users to create their contexts through the application's user interface: manual entering of context data or taking advantage of context recognition. The former basically means that the user interface contains forms etc. through which the user can try to define the context as accurately as he can. This approach may be difficult for the user and also for the designer especially if there are more than a few context elements to define. In the latter approach context information is automatically collected to the user interface through different sensors. The user may then be left to decide which information is relevant in defining that specific context. This approach expects that the user is situated in the context that is being defined. However, if that is the case, it then eases the burden of defining the context, because recognition is known to be much easier than recall. Of course, all information, e.g., user's feelings, may not be possible to input through sensors and that information must in all cases be entered manually.

Probably the simplest form of automatic context activation is to use mobile phone's calendar information. The success of that method depends on how actively the calendar is used and of course it does not cover any spontaneous activities, but nonetheless provides some context information without bothering the user. (Khalil et al., 2005) More elaborate recognition based context definition approaches have been studied for example in the a CAPpella (a Context-Aware Prototyping environment for end users to build applications without writing any code) project. In a CAPpella both the context and the associated action were recognized. This way also the often used rule-based approach for associating contexts with actions was eased for the end user. In a CAPpella the user first recorded the behaviour (situation and action) that he wanted a CAPpella to learn. Available sensors were video camera, microphone, RFID, an indicator of whether the phone was in use, and detectors for actions such as logging in and logging out of a computer, sending email, turning lights on or off etc. After the data had been recorded, the user then selected relevant events from the recording and used them to train a CAPpella by repeating the process a few times over a period of days or weeks and this way improved a CAPpella's ability to recognize the behaviour. After a sufficient number of training examples had been provided, the user could tell a CAPpella to recognize the situation, and when it did, it performed the demonstrated actions. The system was used successfully in two scenarios: to detect when a meeting occurred and to detect whether the user had taken his medicine. (Dey et al., 2004)

Generally, activities involving highly repetitive body motions such as walking and running may result in good recognition capability with only a small amount of user specific training data, but more complex activities like cooking might need a lot more data. In these more complex applications the user interface for training the application must be designed with care, because the inconvenience caused by training the application to recognize different behaviour should not exceed the value that the application provides otherwise users will not find it worth doing. (Intille et al., 2004)

The PePe project (Lehikoinen et al., 2006) studied the sharing of location information and especially what kind of names users would typically use for locations and for what purpose. It concluded that automatically updated location information was the most important context element together with status text and status image. The locations that the participants created were classified in three categories: points of interest, generic locations, and geographical areas. Points of interest meant places generally known by local people like movie theaters, cafés, or shopping malls. Generic locations signified places that could be understood only by people within a certain social network like home, friend's home, school, or work. Geographical areas were locations like cities, districts, or countries. Geographical areas were also sometimes combined with other types of locations, e.g., a café in a certain city especially to distinguish between two similar locations. Generally the names were more specific on familiar areas and more generic in distant locations. In addition, users often interpreted "unknown" location to mean that the person was "on the move" or if the location remained "unknown" for a long period of time, it was considered that the user was not willing to reveal the location. Sometimes users also forgot to name locations, which could indicate a need for the application automatically create or suggest locations for the user whenever possible.

3.3 Representing Context Information

One interesting notation for representing contexts is based on the natural language structure where a simple sentence often takes the form of <subject> <verb> <object>. Respectively, context can be defined as (<ContextType>, <Subject>, <Relater>, <Object>), where ContextType refers to the type of context (e.g., location), Subject is the person, place, or thing with which the context is concerned, Relater is a comparison operator (such as =, >, or <), verb, or preposition that relates the subject and object, and Object is a value associated with the subject. (Ranganathan et al., 2002)

Example contexts presented in this notation include:

- Context(Location, Andy, Entering, Kitchen)
  Andy enters kitchen.
- Context(Social Relationship, Bob, Friend, Chris)
  Bob is Chris's friend.
- Context(Time, Helsinki, Is, 12:00)
  Time is 12:00 in Helsinki.

Though being rather simple, this notation can express most basic context types. It is also independent of the actual implementation. The notation also supports more complex contexts by allowing Boolean operators (conjunction, disjunction, and negation) and quantifications (existential and universal). (Ranganathan et al., 2002)

Examples of more complex contexts using these mathematical operations include:

- Context(Location, Andy, Entering, Kitchen) ∧
  Context(Social Activity, Dinner, In, Kitchen)
  Andy enters kitchen where a dinner is going on.
- Context(Social Relationship, Bob, Friend, Chris) ∨
  Context(Social Relationship, Bob, Friend, Dan)
  or more simply Context(Social Relationship, Bob, Friend, Chris ∨ Dan)
  Bob is Chris's or Dan's friend (or both's).
- ¬Context(Location, Eric, In, Living room)
  Eric is not in the living room.
- ∃Person Y Context(Location, Y, In, Kitchen)
  There is someone in the kitchen.

In practice one has to concentrate on a limited set of context types. Dey et al. (2000) have stated that location, identity, activity, and time are the most important context types. On the other hand, Schilit et al. (1994) say that the most important aspects of context are where are you, who are you with, and what resources are nearby. Dey et al. (2000) have defined context as a fourtuple: (location, activity, time, identities of nearby people), where any element can be a wildcard. Whenever all conditions are met, the context is considered active.

Example contexts in this notation include:

- (At home, *, *, *)
  The person is at home.
- (At work, Programming, 12:00, John)
  The person is at work doing programming, time is 12:00, and John is also nearby.

Context information can also be systematically structured by the creation of an ontology, i.e., a shared understanding of a certain domain, which is typically presented as a set of entities, relations, functions, axioms and instances. Context ontologies are useful as they enable shared understanding of context information between different information providers and collaborating agents. They also make it possible to do context reasoning, i.e., to check the consistencies of contexts and to deduce high-level, implicit context elements from low-level explicit context data. In addition to that, context ontologies facilitate rapid development of applications, more efficient use of resources, as well as reuse. Existing ontologies can be used as basis for developing more complex ontologies without starting from scratch. Examples of developed context ontologies include CONON (Wang et al., 2004) and CoOL (Strang et al., 2003). (Korpipää et al., 2003a)

In CONON the context model is divided into upper ontology and specific ontology. The upper ontology contains general features of basic contextual entities while the specific ontology defines the details of the general concepts in each sub-domain. The main components of the upper ontology are *Location* (*IndoorSpace*, *OutdoorSpace*), *User*, *Activity* (*DeducedActivity*, *ScheduledActivity*), and *ComputationalEntity* (*Service*, *Application*, *Device*, *Network*, *Agent*). All of these then have their own sub-classes which can be

defined to model specific contexts in given environments. For example, *IndoorSpace* in home domain could be classified into subclasses *Building*, *Room*, *Corridor* and *Entry*. All entities are also associated with attributes and relations to other entities. The context model is found feasible and necessary for supporting context modelling and reasoning in pervasive computing environments. (Wang et al., 2004)

In CoOL aspect, scale and context information are the core concepts of the context model. An aspect is a classification, symbol- or value-range, whose subsets are a superset of all reachable states, grouped in one or more related dimensions called scales. A scale is an unordered set of objects defining the range of valid context information. Context information is any information which can be used to characterize the state of an entity concerning a specific aspect. For instance, the aspect "GeographicCoordinateAspect" may have two scales, "WGS84Scale" and "GaussKruegerScale", and a valid context information may be an object instance created in an object-oriented programming language with new GaussKruegerCoordinate("367032", "533074"). (Strang et al., 2003)

Since there are many different ways to represent contexts, one has to choose the most appropriate ones for one's own application development. The key is to look for those representations that are just sufficient for one's purposes. Choosing too complex notations or ontologies can make the application development unnecessarily complicated. Still, one has to think about possible future needs as well and also consider the extensibility of the representations.

3.4 Obtaining Context Information

The ways for users to obtain other persons' context information can be roughly divided into two groups: heavyweight methods and lightweight methods. Heavyweight methods require user's full attention. Examples of these include calling on the phone, sending/reading email, or watching a web camera video from the other person's location. Lightweight methods, on the other hand, do not require user's full attention, but allow other tasks to be executed in parallel. Examples of these include looking out of the window to see what the weather is like, hearing someone's footsteps entering a room, or smelling a familiar perfume. Lightweight methods often take advantage of one or more human senses. (De Guzman et al., 2004)

The way context information is typically conveyed in context-aware software can also be categorized as lightweight as it is usually done by the means of different visual or audible aids such as appearing images, movement or sounds. For example, widely used instant messaging applications like MSN Messenger use sound clips and flashing windows to indicate when a buddy goes online or offline. These applications have become especially popular among teenagers at home and employees in work environments since they allow users to follow their online buddies' statuses and to easily exchange messages, images, documents etc. with each other. However, the indications of context changes can be distracting to users working on some other tasks. Therefore, one design objective

for context-aware applications should be to minimize these distractions while still allowing context changes to be notified. (De Guzman et al., 2004)

De Guzman et al. (2004) also studied the use of instant messaging software with 15 users that were divided into two focus groups. The study showed that the status information is often used to trigger communication through more heavyweight means such as telephone or face-to-face conversation. However, they conclude that this probably depends a lot on what kind of relationship the buddies have and what kind of cultural factors are in effect.

3.5 Privacy Concerns

Privacy can be defined as the right to determine when, how, to what extent, and to whom information about oneself is communicated. It can be divided into information privacy (how our personal information is handled by the government or different organisations), bodily privacy (such as bag searching), privacy of communications (right to communicate without being surveylled), and territorial privacy (right for privacy in our homes etc.). Location privacy can be seen as a subtype of information privacy and it can be defined as the right to prevent other parties from learning one's current or past location. Location privacy is especially important in context-aware applications since location is the most widely used context information. There are different levels on how location information can be used to invade privacy. Two extremes would be to get the name of the city where the located person is at some moment of time or to get the exact location of the person at all times. Obviously the exactness of the location information and the interval at which it is collected have a significant effect to the level of privacy. Most research on privacy has focused on anonymity and secrecy, but they are relevant approaches only when the application is not based on sharing information with others. For example, a teleporting application that utilized user's location and the location of workstations to dynamically map the user interface to whatever computer that was near the user at any time, is an example of an application that could take advantage of anonymity to protect privacy. However, context-aware communications applications are generally based on the sharing of context information with others and anonymity is, therefore, not desired. In context-aware communications participants are usually already aware of each others' identities and the privacy risks have more to do with the need to avoid undesired social obligations or potentially embarrassing situations. (Beresford et al., 2003; Hong et al., 2004)

Barkhuus et al. (2003) studied how people feel about their privacy when their location is being tracked. The study consisted of 16 participants who filled a 5 day-journal by answering pre-specified questions about the usefulness and level of concern in using presented location-based services. However, the services were not implemented, but the participants were asked to "imagine" the existence of the services. The results indicated that the attitudes are more positive if people can opt to turn the tracking off and if the tracking is based on only the device knowing its own position instead of collecting extensive amounts of tracking information to the server. Also, the findings suggested that attitudes

are positive as long as the users perceive the location-based applications to be useful.

People have different relationships between each others. Typically, the amount and accuracy of context information that people are willing to reveal varies based on this relationship. Davis et al. (2005) examined what kind of context information people are willing to reveal and to whom. Their research method consisted of a questionnaire that 16 persons answered. The purpose of the questionnaire was to find out what amount of information from different sources (e.g., video, audio, location, telephone, calendar, computer activity) people would disclose to seven different relationship types (friend, supervisor, peer, subordinate, spouse, secretary, acquaintance). They found out that although people would reveal different amounts of information to different relationship types, the only relationship type that was consistently given less information, was the acquaintances, i.e., people that the users did not have a strong relationship with. People were also willing to reveal more information about their locations and calendars than they were about their computer use, telephone conversations, or office audio.

People are also often more sensitive towards revealing certain context information, while more willing to reveal something else. For example, revealing location may be more comfortable to the users than revealing the persons with whom they are. However, revealing some context information may in some cases lead to the deduction of other information. For example, if a person reveals that s/he is in a movie theatre, one may deduct that the person is most likely with someone and the activity s/he is engaged to is watching a movie. (Khalil et al., 2006)

Often people are not even aware of what they are actually revealing so the application should assist the users in comprehending the aggregate effects of their actions and prevent undesired revelations. However, should the application automatically hide potentially revealing information or should it only alert the users and let them decide how to handle the situation? There are benefits and drawbacks in both approaches. If the application hides the information automatically, it promotes the idea of technology that disappears to the background and lets users concentrate on what they are actually doing. On the other hand, users typically want to be in control of things related to their privacy. (Ludford, 2006)

Not sharing the context information but just allowing messages to be sent to certain contexts is one possibility to support privacy while still allowing context-aware messaging. In this approach the sender would define the recipient's context in which the message should be delivered. For the recipient, the message would seem like any regular message. This method was implemented in DeDe (Defined Delivery) system and studied with seven users. Used context elements were time, location based on cell id, phone call to or from a certain number, and a certain Bluetooth device appearing nearby. The results indicated that this kind of delayed delivery was useful in allowing the messages to be constructed when most appropriate for the sender while still enabling sensitivity in regards to the delivery time. However, it caused a higher cognitive load on the sender, as s/he

had to think not only the present but also the future situation of the recipient when constructing the message. This also required that the sender was somewhat familiar with the recipient's habits and routines so that s/he was able to define the desired delivery context properly. In the study, location and time were clearly the most used context elements, which could indicate that phone calls and Bluetooth proximity might be non-useful or too complex concepts for some users. Delayed messages also posed some design issues, namely the possibility to get delivery reports and to check and possibly modify or even remove outgoing messages. To support privacy delivery reports could be implemented so that the recipient would be asked whether s/he wanted a delivery report to be sent after s/he had read the message. Finally, it must be noted that this kind of approach would only work with messaging applications not with, e.g., phone calls. (Jung et al., 2005)

3.5.1 The Four Aspects of Privacy

There are four key aspects that end users are concerned regarding privacy (Raento et al., 2005a; Hong et al., 2004). These will be presented shortly in the following.

Firstly, the **value** that the users will get when revealing their personal information must be visible to the users. This will enable the users to properly evaluate the trade-off between how much information they are revealing and what are they gaining in return. (Hong et al., 2004)

Secondly, users want **control** over and feedback about what information is visible to whom in a certain situation. However, it would require complex user interfaces to thoroughly model the precise access control mechanisms that users have in their minds and these mechanisms would also be quite dynamic in nature which would require constant interaction with the user. It is hardly ever the case that the value that the user is getting from the application would be enough to compensate the burden of spending a lot of time on defining and updating access control rules. Therefore, simple access control and basic notifications supplemented with the ability to comment and manipulate the automatically collected information are often sufficient. It has been shown that the identity of the inquirer is more important than the situation when deciding whether to reveal context information, although situation is important too, especially if the inquirer is the user's employer (Lederer et al., 2003b). Therefore, access control can be used to primarily select what information each contact is allowed to view. Secondarily, there may also be an option to specify in which context the information should be viewable. Notifications can be used to track what information each contact has looked at. These can be collected into some logging feature of the user interface. (Hong et al., 2004; Raento et al., 2005a; Raento 2007)

Thirdly, **plausible deniability** that is the ability to plausibly deny revealing personal information is also a desired component. It is naturally used for example in cell phone calls, where not answering a call can be because the callee doesn't want to talk to the caller or, e.g., because the callee didn't hear the phone

ringing. The caller cannot know the reason. The requirement for plausible deniability rises from the social need to avoid potentially embarrassing situations, undesired intrusions, and unwanted social obligations. Since most of the context information is collected automatically, there is no simple way to achieve plausible deniability. It has been suggested that, the application should allow the user to select whether real or fake information is shown to the inquirer. Fake information could be inputted manually or automatically generated. The inquirer wouldn't know if the information is real or fake. For example, if one said to his wife that he went out to the wild but instead went to a bar, he could fake the context information to indicate that he indeed is spending time in the nature. (Hong et al., 2004; Raento et al., 2005a)

Fourthly, **reciprocity** that is the need that all parties reveal something about themselves is also needed in many situations. In face-to-face conversations reciprocity is needed to build trust, deepen the relationship and facilitate an understanding of the other person's interests, tastes, needs and desires. In the same sense in certain context-aware applications it would seem sensible that if one is not willing to reveal anything of him, he wouldn't see others' information either. Whether the revealed information should match exactly that is whether A is only allowed to see B's location if A is also willing to reveal his location to B, is another question, since this would somewhat contradict with the ability to control what information is revealed and the ability to fake information. (Raento et al., 2005a; Raento, 2007)

3.5.2 Designing for Privacy

One possible way to design context-aware applications on the privacy viewpoint is to decide whether the application should be pessimistic, optimistic or mixed-initiative. In pessimistic applications end users define beforehand exactly what information and when is revealed to whom. On the other hand, optimistic applications are based on logs and notifications that can be used to detect abuses. Optimistic access control is useful when openness and availability are more important than complete protection and it is also easier to take into use than pessimistic access control, since the user doesn't need to think about all possible situations beforehand. Mixed-initiative mixes pessimistic and optimistic approaches in such that in it, users are interrupted and given a choice whether to reveal information, when someone requests the information. For example, choosing whether to answer a phone call based on the identity of the caller is an example of a mixed-initiative approach. (Hong et al., 2004)

Lederer et al. (2003a) have designed a user interface for defining access control rules for contexts based on the central notion that people disclose different versions of personal information to different parties under different conditions. The user interface is divided into three areas. Firstly, there are the inquirers, e.g., user's friends or other contacts. Secondly, there are situations that are the context information of the user. In the example user interface they use a fourtuple (location, activity, time, identities of nearby people), but basically situation could contain any elements of context information. Thirdly, there are faces, which define what information and how precisely is revealed to the inquirers in all

situations. They conclude that their approach is superior to simple, automated disclosure strategies, but can be simplified further still. It would be especially useful to consider static and dynamic context information separately. People that the users have a relationship with, already know the static information, but revealing dynamic information to them might be more sensitive than revealing dynamic information to strangers. Also, the concepts of situation and face might be a bit confusing especially when dynamic context information is considered as situation presents the actual context and face the transformed version that the user wishes to convey.

In general, configuring privacy permissions at a group level seems to provide the best balance between privacy control and the effort of configuration as the study conducted by Patil et al. (2005) shows. Also, default values and templates could be used to further ease the burden of creating the configurations. However, it is essential to set the defaults right, since most users are not likely to modify them.

Also, to increase usability it should be possible to re-use the users' privacy preferences of one application in other similar applications. It is however a challenge to reuse the preferences across different organizations and application areas. Nonetheless, it should be possible to at least define preferences for a certain category of applications (e.g., telecommunications or web-commerce). (Hull et al., 2004)

All in all, privacy issues in context-aware communication are about restricting who can see what information and in which situations. It is a challenge to create a simple yet powerful enough user interface for this purpose. Group level permissions, default values, templates, and re-using preferences in different applications should be utilized extensively.

3.6 Implemented Context-Aware Communication Applications

This chapter presents implemented context-aware communication applications and prototypes, which include Kontti, ContextContacts (Jaiku), iCAMS, InfoRadar, Live Contacts, Socialight, and Context Watcher. Kontti and ContextContacts (Jaiku) are analysed in more detail while the features of the other applications are introduced more shortly.

3.6.1 Kontti

Kontti (Kolari et al., 2004) was the result of a two-year project that took place at VTT in 2002-2003. Its goal was to develop concepts and tools for offering context-aware mobile services. In the project a context-aware service platform was developed. The platform provides personal management and sharing of contexts and presence information, content adaptation, and context-aware messaging. Contextual information can be viewed and managed within the system. The main screen of Kontti is presented in Figure 2.

Figure 2. Main screen of Kontti.

Goals of context-aware communication applications were discussed earlier. Right message at the right time was one goal that was recognized. In Kontti this is enabled by allowing the sender to define that the message is to be sent only after the recipient is in a certain context. However, there is no option to define that the message is to be sent after the sender is in a certain context. Kontti also supports the goal of sharing awareness of one's context.

It was discussed earlier that according to Dey et. al (1999), location, activity, time, and identities of nearby people are the most important context types. In Kontti the used context elements are location, time interval, and activity. Therefore, of the most important context elements only the identities of nearby people is not utilized.

Kontti takes advantage of network operator's positioning service via the LIF protocol and also WLAN positioning for getting the location information, but regards to other context information, Kontti mainly relies on manual entering of context data. Context changes are not visibly indicated in Kontti and privacy issues are mainly tackled by allowing users to define what context information is public to whom.

Kontti was evaluated with several field trials and surveys with a total of 98 interviewees and 131 survey respondents. Results provided several interesting findings. The need to integrate the features with the mobile device's own operating system arose as users have gotten used to sending messages using the mobile device's own messaging application. In addition to that, users stated that changing one's own context information should be either automatic or as easy as changing the mobile device's profile. The possibility to add audiovisual context information, like photos or media clips, was also brought up to allow more creativity and to emphasize the role of presence as the communication channel.

The business potential of Kontti was analyzed by interviews with several service providers. Most business potential was found for companies that focus on arranging events. It was also highlighted that the service providers should have a proper content management tool for updating the service content and the end users should be provided with easy-to-use user interfaces in order to make it effortless to start using the services.

The following open research questions were left by Kontti:

- How to represent context ontology in a way that is suitable for the requirements set by context-aware services?
- How to handle different location techniques so that they complement each others?
- How to use RFID tags to identify contexts?
- How to use the mobile phone's profile settings to assist in recognizing the context?
- How to use the recognized context to change the mobile phone's profile settings?
- How to adapt content based on context?
- How to seamlessly synchronize data between mobile phone's local storage and external repositories?

3.6.2 ContextContacts (Jaiku)

ContextContacts (Oulasvirta et al., 2005) was built on top of the ContextPhone (Raento et al., 2005b) platform running on Nokia series 60 mobile phones. ContextPhone and ContextContacts were developed in 2002-2005 by the Department of Computer Science and HIIT Basic Research Unit, both at University of Helsinki, and of HIIT Advanced Research Unit. ContextPhone's goals were to study what are the users' understandings of their current contexts, how to make automatic inferences about the contexts, and how to characterize context to users and design user interaction about contexts. ContextContacts concentrated on re-designing mobile phone's contact book to provide cues of the current situations of others therefore supporting mobile communication decisions and group coordinations. Example screens of the ContextContacts application are presented in Figure 3.

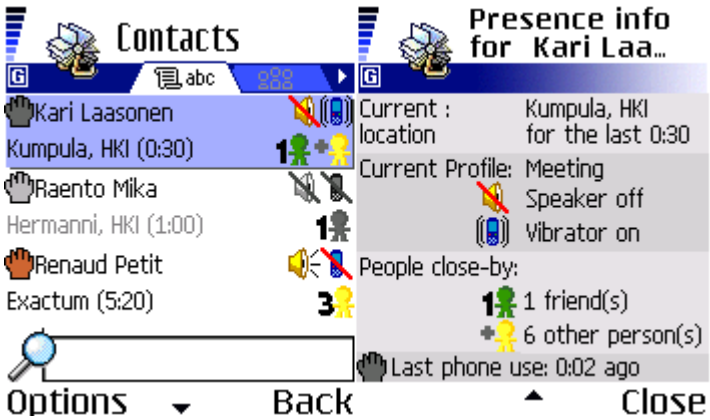

Figure 3. Contacts and presence info screens of ContextPhone.

ContextContacts does not support context-aware messaging, but it concentrates on sharing the awareness of one's context. For this purpose it uses current location, current mobile phone profile (including speaker and vibrator status), last time phone was used, and people close-by.

As one can see, current location and people close by map directly to the definition of Dey et al. (2000). Current mobile phone profile can be seen to map to the activity in the definition since it can be used to somewhat deduce the current activity. For example, the mobile phone profile “meeting” indicates that the person is in an environment where he wants to be discrete whether it is an actual meeting or not. Also, last time phone was used maps somewhat to the time in the definition. Naturally, the current time as context information is not needed in the application since it is universal in nature.

What context information is selected to be used in real-life applications like ContextContacts has obviously much to do with the availability of different sensors. ContextContacts uses network operator’s positioning service as well as GPS for getting the current location. It also uses Bluetooth scanning for discovering people close-by and built-in features of the phone to track current mobile phone profile. For other context information, ContextContacts also mainly relies on manual entering of context data. However, it does use network operator’s automatic identification and naming of important locations from the logged cell data, which in a way automates the context creation.

In ContextContacts the context changes are not visibly indicated either and privacy issues are also dealt with by allowing users to define what context information is public to whom. Neither plausible deniability nor reciprocity is implemented in practice.

ContextContacts was developed into a commercial service called Jaiku (http://www.jaiku.com/), which was published in July 2006. Figure 4 illustrates the functionality. Nowadays, Jaiku is a growing social networking and micro-blogging service which utilizes both web and mobile phones. It also allows programmers to make their own third party software components through its public API. In June 2007 there were about 40 000 registered users. In October 2007 Jaiku was acquired by Google (Google, 2007).

Figure 4. Jaiku.

3.6.3 iCAMS

iCAMS (Nakanishi et al., 2004) is a context-aware messaging service that uses location and schedule information. It was in the University of Electro-Communications in Tokyo. In iCAMS the location information is automatically generated using NTT DoCoMo's location-detection service and it can detect locations within approximately 100 meters. The user interface allows sorting of available communication channels based on schedule and location.

The system was evaluated with two groups of ten users in eight week user studies. Results indicate that users find location and schedule information useful for initiating communication but also just for obtaining information of other users. For the latter purpose, a need to see which users were together arose. Sometimes the users also chose the appropriate medium to be face-to-face, if they saw that the person they wanted to communicate with was close-by. Most users found the 100 meter location radius too imprecise. It was especially brought up that the accuracy of the location should change based on whether the area was small (e.g., indoors) or large (e.g., outside). If the users had knowledge of each other's behavior they could however conclude each other's locations rather well even though the provided location information was quite inaccurate. General opinion was that privacy issues would not be a problem when the system was used among friends, families, or colleagues, who are peers. However, if the system was used among hierarchically related people, the possibility to use the tool for monitoring instead of communication brought up some concerns.

3.6.4 InfoRadar

InfoRadar (Rantanen et al., 2004) implements location-based messaging. It provides a novel radar interface for seeing where the messages are located. The radar interface makes the user feel that s/he is in a mixed-reality space much better than a simple list-based user interface would do. It is also applicable everywhere whereas a map-based user interface is only applicable in mapped areas. The scanning radius can be varied from close proximity up to 12 kilometres. The messages can also be read elsewhere than in the actual location where they were sent. The system also supports filtering messages based on categories, multimedia messaging, social activity indicators, and voting. Filtering messages prevents the screen from getting clustered with messages. Multimedia messaging allows capturing the physical context in a much richer way than using only text-based messages. Social activity indicators mean that the system displays traces of users' movements to give a sense of social activity in locations. This can motivate users to post messages since they know that there is potential audience. Voting allows the inquiry of public opinion from a large population, which can also indirectly trigger discussion and activity elsewhere in the system.

The system was tested in three week field trials with two groups that both consisted of six people. In one group the members knew each other from before and in the other group they were previously unknown to each other, but worked

or did business in the same location (a shopping mall). Both groups took advantage of location-based messages, but the previously unknown people used those features somewhat more. This could indicate that location-based messaging may be especially useful in engaging social interaction within unknown people. One interesting finding was that people expected locations to be quite accurate, e.g., just outside a certain shop. Perhaps, because of this the radar interface generated mixed feelings among the users. Some liked it, and some found it confusing in determining the exact location. One example was its unsuitability to display message locations in a shopping mall that had multiple floors. The voting functionality was also found quite useful, e.g., for joking and planning.

3.6.5 Live Contacts

Live Contacts (Ter Hofte et al., 2004) emphasizes on sharing calendar information between persons. The calendar information is automatically extracted from the mobile device's calendar application. It includes both current and near-future calendar appointments. Live Contacts also supports instant messaging status and location information as context elements and it shows availability preferences (red, orange, green) for communication media (work phone, mobile phone, home phone, SMS, Messenger, e-mail) while allowing the user to make contact with the press of a key either immediately or later via a reminder. Live Contacts has both a mobile client and a desktop client. The mobile client is illustrated in Figure 5.

Figure 5. Live Contacts.

3.6.6 Socialight

Socialight (Melinger et al., 2004) started as a research project. Its original functionality consists of notifying users of nearby friends and of using digital gestures and location-based messages for communication. The server tracks the locations of all users and notifies them when their contacts or contacts' contacts are within a specified distance. The users can then view information of the nearby users or initiate communication with them. Digital gestures mean that the users can send vibrations of various lengths to other persons' phones. It is a light way of communication which does not require much concentration. It can be seen to somewhat correspond to the physical world's communication of glancing or tapping someone on the shoulder. Possible uses include saying "hi" to a friend, being flirtatious, or giving a virtual kick under the table. Location-based messages are messages that are left in geographical places for friends or groups of friends. In Socialight these messages were originally called Sticky Shadows. They are constructed of location, optional expiry time, recipients, and content, which can contain any multimedia elements. When the recipients enter an area where they have messages, they are immediately notified and given a possibility to respond to the messages. Possible uses include life bookmarks (e.g., "buy milk" outside a grocery store), scavenger hunts and games, touring an urban tour, and personal restaurant reviews for friends. Socialight is implemented using Java programming language and it contains user interfaces both for the web and for the mobile device. Location is tracked using Bluetooth, GPS, or network operator data. Nowadays, Socialight has developed into a commercially available free service which concentrates only on location-based messaging. The term Sticky Notes instead of Sticky Shadows is now used for the geo-tagged messages. It also has some new features like channels, where users can leave location-based messages for other users interested in a certain topic. Example user interface is presented in Figure 6.

Figure 6. Socialight.

3.6.7 Context Watcher

Context Watcher (Koolwaij et al., 2006) was built on top of MobiLife (Floreen et al., 2005), a generic framework enabling context discovery, exchange and reasoning. Context Watcher enables the sharing of location (based on GPS and/or GSM cell), mood (based on user input), activity (based on reasoning), heart rate (based on heart sensor), speed (based on foot sensor), weather information (based on a location-inferred remote weather CP), and visual data (pictures enhanced with contextual data). The gathered context information can be used in many ways including the following.

- To share awareness of each other's contexts in order to keep in touch with others without having to approach them directly
- As input parameters for information services (e.g., local weather with one click or easy public transportation info)
- Remote logging of activities and preferences and sharing the information with different services like Flickr.com or personal blogs

The mobile application runs on Nokia Series 60 phones. Example user interface is presented in Figure 7.

Figure 7. Context Watcher.

3.7 Conclusions of Related Research

The ways that context information can be used to enhance communications have been quite extensively recognised and examined in the previous research. These are shortly summarised in the following.

Determining the right medium and time for communication has been the topic of many studies (Dabbish et al., 2004; Danninger et al., 2005; Danninger et al., 2006; Fogarty et al., 2005; Gross et al., 2006; Jung et al., 2005; Lei et al., 2004; Malkin et al., 2006; Lei et al., 2002; Ranganathan et al., 2003; Schilit et al., 2002). The issue has been studied both from the contacter's and from the contactee's viewpoint.

Many researchers have also focused on location-based reminders (Dey et al., 2000; Ludford et al., 2006; Marmasse, 1999; Melinger et al., 2004; Rantanen et al., 2004; Schilit et al., 2002; Zhou et al., 2005) while reminders based on other context information have mostly been studied in certain specific environments like medical applications (Miller et al. 2004; Pentland, 2004). There is probably more room for research left in these reminders that utilize more context information than just location. However, this issue is only briefly touched in this thesis as sending reminder messages to oneself to certain situations is one scenario examined in the user survey.

Sharing awareness of one's context is probably the mostly used form of context-aware communication nowadays as instant messengers are commonly used communication tools and services like Jaiku (http://www.jaiku.com/) are becoming more and more popular. Many researchers have focused on what information could be shared (Myles et al., 2003; Nagel et al., 2001; Nakanish et al., 2000; Ranganathan et al., 2002; Schilit et al., 2002 Schmidt et al., 2000; Schmidt et al., 2001; Tamminen et al., 2004). However, not so much research has gone on how the shared information is actually utilized (Fogarty et al., 2004; Paciga et al., 2005). Therefore, concentrating more on this side of the issue might be a fruitful starting point for further research. This topic is however not studied in this thesis.

Finally, reducing ambiguity was the last recognized way to use context information to enhance communication. (Ranganathan et al., 2002; Mankoff et al., 2005) It has not been as big of a research issue as the above-mentioned topics, but nonetheless it can help the communication significantly by making it easier for people to understand each other.

The research on how to create context information has mainly focused on automatic context recognition techniques (Dey et al., 2004; Intille et al., 2004; Khalil et al., 2005). There may be some room for studying what would be the best way to enter context information manually. This thesis tries to find out how important the users actually consider the automatic context recognition compared to the manual input of context data.

Several notations have been suggested for representing and logically combining context information (Dey et al., 2000; Ranganathan et al., 2002). Also, researchers have focused on creating context ontologies to make application development more efficient and coherent (Korpipää et al., 2003a; Strang et al., 2003; Wang et al., 2004). The use of ready-made ontologies is quite sensible when creating more complicated context-aware applications. However, the application framework developed in this thesis only uses location and time as context elements so it would overcomplicate things to use some kind of an ontology in its development.

How changing contexts should be notified has also been studied (De Guzman et al., 2004). The biggest question is how to notify the users without distracting them too much. In this thesis the way that users want to be notified is examined through the user survey.

Privacy issues have been studied as regards to what information users are willing to reveal (Barkhuus et al., 2003; Beresford et al., 2003; Davis et al., 2005; Hong et al., 2004; Khalil et al., 2006; Lederer et al., 2003b; Ludford, 2006; Raento et al., 2005a) and how it should be taken into account when developing the applications and their user interfaces (Hong et al., 2004; Hull et al., 2004; Lederer et al., 2003a; Patil et al., 2005). This thesis will not concentrate on the privacy issues as there are already theses (Raento, 2007) that focus merely on them. However, the users' concerns on these matters are somewhat inspected in the user survey.

Several prototypes that focused on some specific aspect(s) of context-aware communication were also presented earlier (Kolari et al., 2004; Koolwaij et al., 2006; Melinger et al., 2004; Nakanishi et al., 2004; Oulasvirta et al., 2005; Rantanen et al., 2004; Ter Hofte et al., 2004). These prototypes implemented some context-aware communication features and evaluated them with users to find out their potential benefits and restrictions. The aspects that the above-mentioned prototypes studied are summarized in Table 1.

Table 1. Context-aware communication prototypes and the aspects of context-aware communication that they concentrated on.

Prototype	**Aspect**
Kontti	Delivering of messages based on recipients' contexts
ContextContacts	Sharing awareness of one's context
iCAMS	Choosing appropriate communication medium based on context
InfoRadar	Leaving messages on locations
Live Contacts	Sharing calendar information
Socialight	Digital gestures and location-based messaging
Context Watcher	Sharing awareness of one's context

Although the development of an application framework is part of this thesis, the main contribution that this thesis will provide to the already existing research will be a thorough examination of users' needs and concerns on context-aware communication through the analysis of idea movement's ideas and the user survey. In addition to that, the application framework will raise some issues that the application developers should take into account when creating context-aware communication applications.

4. Research Focus

In this chapter the focus of this thesis is refined based on the related research presented in the previous chapter. At first communication is categorized based on different attributes and the attributes relevant to my research are recognized. These attributes are then used to adjust the research questions. After that, different context-aware communication scenarios that this thesis focuses on are extracted from the related research.

4.1. Concepts

I begin focusing my research by categorizing communication based on the following attributes: form (vocal vs. textual), distance (remote vs. face-to-face), electronicity (electronic vs. non-electronic), publicity (public vs. private), and immediacy (instant vs. delayed). There are 5! = 120 ways of combining these attributes and each of them defines some form of communication. Although, it can be said that a small number of combinations dominate the way people usually communicate. Examples of typical combinations are given in Table 2. The focus of my research is also presented in emphasis in the table.

Table 2. Different forms of communication.

Examples	**Form**	**Distance**	**Electronicity**	**Publicity**	**Immediacy**
Electronic message board, mailing list	*textual*	*remote*	*electronic*	*public*	*delayed*
Text-based chat	*textual*	*remote*	*electronic*	*public / private*	*instant*
E-mail, SMS, MMS	*textual*	*remote*	*electronic*	*private*	*delayed*
Traditional message board	textual	remote	non-electronic	public	delayed
Writing letters, leaving notes at home for family members	textual	remote	non-electronic	private	delayed
Nodding, shrugging, using sign language	textual	face-to-face	non-electronic	private	instant
Videoconference	vocal	remote	electronic	public	instant
Phone call	vocal	remote	electronic	private	instant
Yelling	vocal	remote	non-electronic	public / private	instant
Conventional chat	vocal	face-to-face	non-electronic	public / private	instant
Whispering	vocal	face-to-face	non-electronic	private	instant

It can be seen from the table that form, distance, and electronicity are the common attributes for the focus of my research. Therefore, the focus of my research can be narrowed to *textual, remote, and electronic communications*. Examples of such communication are electronic message boards, mailing lists, text-based chats, e-mails, and SMS and MMS messages.

Furthermore, I will only concentrate on those context elements that are *currently supported* by mobile devices either automatically or manually. These context elements include the most important context elements, which were defined by Dey et al. (2000) as location, time, activity and identities of nearby people. Location can be specified for example through GPS receivers either as interconnected or integrated devices or using network operator's positioning service, or WLAN positioning. Specifying time is self-evident. Activity can be specified at least manually. For example Bluetooth can be utilized to recognize the identities of nearby people.

4.2. Refined Research Questions

Based on the focusing described above, the research questions introduced earlier can be adjusted to the following.

- How can currently supported context information be used in a textual, remote, and electronic communications scope?
- What advantages and concerns there are in using currently supported context information in a textual, remote, and electronic communications scope?

Therefore, the refined research questions are a subset of the original research questions as only currently supported context information is considered and instead of all mobile communications the scope is textual, remote and electronic communications.

The first research question covers both technical possibilities as well as users' needs. However, the emphasis is on the technical issues and the question is therefore answered mostly based on the related research. The second question concentrates more on users' viewpoints and is answered mainly based on the empirical research.

4.3. Scenarios

In general, scenarios are descriptions of how a system is used. They are exploited in various purposes, like requirements gathering, marketing, or testing. The related research presented in the previous chapter contained several implicit scenarios that are essential in context-aware communication. In the following, I will formulate these scenarios explicitly as features of a context-aware system. In my research, I will focus on these scenarios and examine their possibilities, advantages and concerns with users.

- Seeing other person's current context information (*3.1.3 Sharing Awareness of One's Context* and *3.6 Implemented Context-Aware Communication Applications*)
- Restricting what context information about you other persons are allowed to see in different situations (*3.5.1 The Four Aspects of Privacy*)
- Seeing what context information about you has been sent to others and when (*3.5.1 The Four Aspects of Privacy*)
- Automatic activation of your own context information instead of having to manually change them every time your context changes (*3.2 Creating Context Information*)
- Notification as a sound, small icon etc. every time your automatically activated context changes (*3.4 Obtaining Context Information*)
- Notification as a sound, small icon etc. every time a certain person's context changes (*3.4 Obtaining Context Information*)
- Sending of messages so that they are delivered to the recipient only after the sender is in a certain context (*3.1.1 Right Message at the Right Time* and *3.6.1 Kontti*)
- Sending of messages so that they are delivered to the recipient only after the recipient is in a certain context (*3.1.1 Right Message at the Right Time* and *3.6.1 Kontti*)
- Cancelling or modifying messages before sender and/or recipient have been in such contexts that the messages could have been sent (*3.1.1 Right Message at the Right Time*)
- Automatic sending of messages repeatedly every time the sender and/or recipient arrive at certain situations (*3.1.1 Right Message at the Right Time* and *3.6.1 Kontti*)
- Seeing whether the message has already been delivered to the recipient and whether the recipient has already read the message (*3.1.1 Right Message at the Right Time*)
- Sending messages to oneself to certain situations, e.g., as reminders (*3.1.2 Reminders Only when They Are Actually Relevant*)
- Leaving messages to certain places for anyone that arrives at the same place to read (*3.6.4 InfoRadar* and *3.6.6 Socialight*)

In the development of the application framework for context-aware messaging I will concentrate on investigating the scenario of leaving messages to certain places for anyone that arrives at the same place to read.

5. Results

This chapter presents the results of the analysis of the idea movement's ideas, the web questionnaire, and the creation of an application framework for mobile context-based messaging.

5.1 Analysis of the Idea Movement's Ideas

In spring 2006 VTT Technical Research Centre of Finland collected 35 000 ideas for new mobile services from people in all age groups using a number of brainstorming workshops around Finland (Alahuhta et al., 2006). I inspected 4000 of these ideas to find those that relate to the focus of my research. Half of the analyzed ideas were from the University of Oulu and the other half from the adult/occupational upper secondary school of Rovaniemi. All the ideas were originally written in Finnish. Table 3 presents how different context elements were present in the ideas. The context elements were chosen based on what kind of ideas there seemed to be. Therefore, emotion / mood is also included in addition to location, time, activity, and identities of nearby people.

Table 3. The presence of different context elements in the idea movement's ideas.

Category	Description	Example	Share
Location	Location of the user or some object.	Locating the nearest restaurant.	25 %
Time	Time that is relevant in a non-trivial way.	Retrieving the weather forecast for a specific moment.	4,7 %
Activity	What the user is doing.	Searching for services based on the current activity.	1,8 %
Identities of nearby people	Identities of people that are close-by.	Recognizing likeminded people that are near the user.	1,8 %
Emotion / mood	How the user or some other parties feel.	A vigour state analysator that alerts if the user starts to become tired.	1,6 %

The following sections describe in more detail the possibilities that were found regarding the use of context information in a communications scope.

5.1.1 Retrieving Personal Context Information

Being able to see the contacts' locations is an idea that repeats often. Mostly people want to see their friends' and family members' locations. Children are also often explicitly mentioned. The privacy issues are frequently taken into account in a way that it is mentioned that the followed party has to agree to the location tracking. The needs for the accuracy of the location tracking vary from a rough description (at home or out on the town) to more precise information (in which room at home). It is also mentioned that both the initiator and the recipient

of the communication need to see each other's locations. Especially, the called should also be enabled to see the caller's location.

Another interesting context information seems to be what the contacts are currently doing and especially whether they currently or in the near future have spare time. People also want to know if their contacts have similar intentions than they have (e.g., go to a cafe, bar, restaurant, or movie theatre) and if they could join them. Also, a few ideas are about being able to see the contacts' calendar information.

The ability to see in which mood their contacts are also comes up a few times. For example, if one sees that his buddy is depressed, one could try to cheer him up. Other various context information that are mentioned include sizes of clothes, current clothing, allergies, etc.

5.1.2 Context-Aware Messaging

The possibility to send messages that get transmitted on specified times is suggested a couple of times. These ideas include:

- Automatic notification to the boss if one is late from work
- Automatic notification if one is late from a meeting
- Automatic notification to home if one has to work overtime
- Automatic notification in a workplace if someone makes coffee
- Automatic notification to certain persons when one is home and available
- Automatic reminders of important matters to family members
- Automatic notification to parents if children leave the house or some defined border
- Automatic notification if a friend is close by
- Automatic reminder when one needs to go some place
- Automatic reminder when school assignments need to be returned

5.1.3 Context-Aware Chatting

Enriching chatting with context information also comes up. Some suggest directly a MSN Messenger type application while others suggest more advanced enhancements. For example, seeing the visual appearance (clothing, hair style, etc.) of the communications partner is suggested either through virtual reality or through videoconference. The idea of simultaneous interpreter is also proposed. It could be applied to both written and spoken language. The implementations of these ideas would demand advanced technical features from the mobile device.

5.1.4 Context-Aware Message Board

There are also a lot of ideas for a message board type application. The suggested ideas are basically chatting were the messages may be stored for a period of time and the attendees may be chosen based on context (location) instead of an existing relationship. Suggested ideas include:

- A shopping list that is shared between family members
- A gift list for a person that is shared between his friends and family members
- A message board for the residents of a building
- A ridesharing message board where people who travel to the same direction can find each other and share a car
- Offering services in a neighbourhood, e.g., lawn mowing, taking a dog for a walk, or childminding
- Arranging play dates between families who live in the same area and have children of the same age
- Searching for jogging, sports, dating, etc. company in the neighbourhood
- Sharing user experiences of a certain product, service, place, etc.
- Communicating the presence of a TV licence inspector or police radar in the neighbourhood

5.2 Web Questionnaire

The thirteen different scenarios of context-aware communication that arose from the literary research were studied with 48 users by the use of a web questionnaire (see Appendix A). The purpose was to examine how useful these different scenarios are considered by the users and what concerns and advantages users see in them. Web questionnaire was chosen as the research method since feedback was wanted efficiently from a relatively large number of people before any prototypes had been implemented. The questionnaire was designed to collect both quantitative data (as the usefulness of the scenarios was evaluated on a scale of 1-4) and qualitative data (as respondents could write their comments on each scenario). Research methods and results of the analysis are presented in more detail in Paper I. As a summary, the results indicate user preferences and provide many practical examples of utilizing the scenarios. Furthermore, the results are summed up in the paper to provide practical guidelines for the creation of context-aware communication applications.

The usefulness of the scenarios is further examined here based on age, gender, text messaging activeness, multimedia messaging activeness, and messaging activeness in general. In order to simplify the comparison, age and messaging activeness groups are combined in the following way. Age is divided into groups: “Under 30” (N=27) and “Over 30” (N=21), text messaging activeness is divided into groups: “Sends text messages weekly or more rarely” (N=20) and “Sends text messages daily” (N=28), multimedia messaging activeness is divided into groups: “Sends multimedia messages never or rarely” (N=36) and “Sends multimedia messages once a week or more often” (N=12), and messaging activeness is divided into groups: “Sends messages only sometimes” (N=22) and “Sends messages often” (N=26). The grouping of the messaging activeness is achieved by taking the average of the text messaging activeness and multimedia messaging activeness and using the middle as the division point. The comparison figures are shown in Appendix B.

It seems that young people appreciate their privacy (scenarios 2,3) more while older people are more interested in seeing other people’s context information

(scenario 8). Young people also seem to have more appreciation for the automatic activation of their context information (scenario 4). Other than that, there are no noticeable differences between the two age groups.

The sample size for women is only 13, which puts some doubts in the comparison of genders. However, it would indicate that women are more concerned about restricting who can see their context information (scenario 2) while men are more interested in making the use of the application as easy as possible (scenarios 4, 7, 11, 13). Women also seem to find more use for the cancellation or modification of messages (scenario 5) and to the sending of messages after the recipient is in a certain context (scenario 6) whereas men have more appreciation for delivery reports (scenario 1) and seeing what context information about them has been sent (scenario 3).

The messaging activeness seems to affect the results the most. In general, those that send a lot of messages also found these scenarios more useful. The most distinguished cases are automatic activation of context information (scenario 4), cancelling or modifying messages (scenario 5), and sending of messages after the recipient and/or sender are in a certain context (scenarios 6, 10).

5.3 Application Framework

Application framework for mobile context-based messaging applications was constructed to implement the scenario of leaving messages to certain places for anyone that arrives at the same place to read. The framework provides a way to create messaging applications easily and efficiently by allowing applications to be defined via XML documents. The application framework was implemented using Java programming language. It supports applications that have different categories from which users can search messages based on different context and content criteria and to which users can send messages containing any multimedia elements. Currently supported context criteria are location and time, and the only currently supported content criterion is the words contained in the messages. The architecture of the application framework is illustrated in Figure 8.

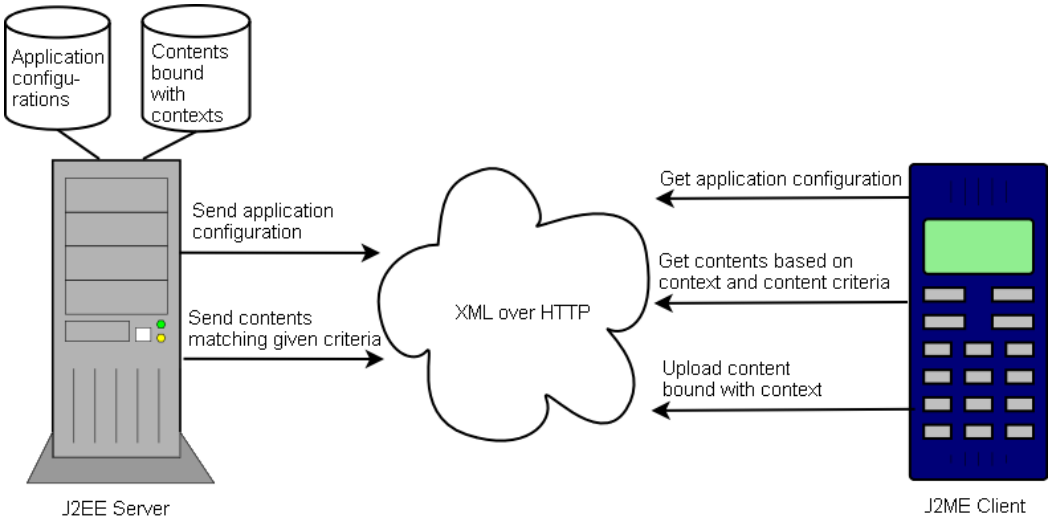

Figure 8. Architecture of the application framework.

The architecture consists of J2ME clients and a J2EE server where the application configurations and contents bound with contexts are stored. These communicate using XML over HTTP. The figure shows mobile clients' typical requests together with the server's typical responses.

Possible applications include reporting and discussing problems of public spaces, rating public services, and selling second-hand products. The main benefit for the users is that it is easy to switch between applications that use the same framework since the applications share a similar look and feel. The framework is presented in more detail in Paper II. An example application that uses the framework is illustrated in Appendix C.

6. Analysis

In this chapter the results of the different research methods are analysed to find out how they link together and what issues they raise regarding the scope of the research.

Analysis of the idea movement's ideas showed that it is easiest to invent location-based services, but there are also possibilities for utilizing other context information. The importance of location information also showed up in the comments on the user survey and in the developed application framework, which implemented location-based messaging. These results strengthen the impression of location being the most important context information.

While seeing other person's context information (especially location, activity, and mood) was present in many of the idea movement's ideas, it was not considered amongst the most useful scenarios in the user survey as seeing other person's current context information was ranked only 8th out of the 13 scenarios. The seven scenarios ranked higher in the user survey where mostly about protecting privacy by restricting what context information was shown and making the use of the application as comfortable as possible by automatically activating contexts and getting notifications of context changes. Therefore, it seems that it is not enough for the application to provide ways to view other persons' context information if privacy and usability issues are not dealt with properly.

One interesting finding is that not only the context information of the person that is called should be displayed to the caller, but also the called person should be allowed to see the caller's context. This turned up both in the idea movement's ideas as well as in the questionnaire. However, the purposes for the caller to see the callee's context and for the callee to see the caller's context are somewhat different. The caller typically wants to see the callee's context to be able to determine, whether the callee is available for communication. On the other hand, as the caller is the one making the decision of whether to initiate communication, s/he most likely is already situated in an appropriate context. The reasons for the callee to see the caller's context have therefore more to do with determining whether the caller can answer certain questions or what kind of discussion topics are socially acceptable.

Several reasons for seeing the other person's context information came up both in the idea movement's ideas and in the questionnaire. However, whether it was determining if the other person was reachable, knowing how far the other person was from an agreed meeting place or knowing for certain where one's children were, the general goal was almost always removing uncertainty. Only one other generic goal could be recognised as finding out that a buddy was depressed and cheering him/her up was more about helping others than removing one's own uncertainties.

The idea movement's ideas also included several scenarios where the messages were sent automatically on specified times. Examples of these were automatic

notifications to the boss if one was late from work and automatic notifications to home if one had to work overtime. However, in the user survey the scenario of automatically sending messages repeatedly every time the sender and/or recipient arrived at certain situations was considered the least important. This may be because the concept of context-based automatic messaging is not that self-explanatory at first. If the user survey had included an example of such a scenario, perhaps the users would have valued it more. On the other hand, an example scenario could have steered the users understanding of the scenario, which was not desired either. Nonetheless, it can be concluded that automatic context-based messaging is not an easy concept to comprehend and although it was not valuated highly in the user survey, it may still have some potential which deserves further studying.

Several suggestions for a message board type application were also present in the idea movement's ideas although leaving messages to certain places for anyone that arrives at the same place to read was ranked only 9th in the user survey. The same reasoning that was given earlier applies also here. The scenarios ranked higher in the user survey where mostly about protecting privacy and easing the use of the application which users seemed to appreciate the most. The developed application framework could be utilized to implement many of the suggestions for a message board type application including (1) a message board for the residents of a building, (2) a ridesharing message board where people who travel to the same direction can find each other and share a car, (3) offering services in a neighbourhood, e.g., lawn mowing, taking a dog for a walk, or childminding, (4) arranging play dates between families who live in the same area and have children of the same age, (5) searching for jogging, game, sports, dating, etc. company in the neighbourhood, (6) sharing user experiences of a certain product, service, place, etc., and (7) communicating the presence of a TV licence inspector or police radar in the neighbourhood. Some of these (3, 4, 5, 6, 7), could use predefined categories while others (1, 2) might take advantage of dynamic, user-specified categories. Most of these (1, 2, 3, 4, 5, 7) could take advantage of location information and time information would be relevant in all of them. A possibility to leave the messages only for a certain short period of time and letting the messages get automatically deleted after that could be implemented in the application framework and utilized in some of the scenarios (2, 3, 4, 7). Also, in some of the scenarios (2, 3, 4) it would be useful to implement a functionality of pushing the most topical messages forward at regular intervals.

The application framework also highlighted the problem of deciding the moment at which to bind the context to the message, i.e., if there are many content elements in the message should the context be bound to each element individually or only once to the whole message and if the latter, at which moment should the binding be done. This would also have to be considered in many of the idea movement's ideas, especially those about context-aware message boards. It would not be easy to find a balance between the accuracy of the context and the amount of data generated. Furthermore, mobile devices would pose significant limitations for generating huge amounts of data as they have limited memory and processing capabilities compared to, e.g., desktop computers and laptops. Also, displaying the data in the relatively small user

interface of a mobile device in a usable way would be a challenging task. In addition to that, privacy issues should be taken into account, if the context was to be tracked continuously. One would also have to consider how the context was defined if the application allowed messages to be constructed through a web interface at a desktop computer as in that case the context at which the message was sent would not probably be the best choice if it could be defined at all.

7. Conclusions

In this thesis, a literature study on context-aware communication was presented. Different ideas that relate to that scope were extracted and analysed from the idea movement's ideas. Also, the thirteen context-aware communication scenarios that were recognized from the study were evaluated with users using a questionnaire. In addition to that, a mobile context-based messaging application framework was developed.

7.1 Discussion

Next, the used research methods are discussed separately.

7.1.1 Idea Movement's Ideas

Analysis of the idea movement's ideas revealed several interesting ideas related to context-aware communication. Location information was by far the most needed feature for actually being able to implement the ideas. Also, there were some ideas that could utilize Bluetooth or NFC. The ideas themselves were not that different from the ideas that arose from the questionnaire. For example, location-based message boards were suggested in both. This indicates that there really is need for these kinds of applications.

As only 4000 out of 35 000 ideas were inspected, one could question whether the results would have been different had more ideas been examined. There is of course no definite answer. However, the general categories (retrieving personal context information, context-aware messaging, context-aware chatting, and context-aware message board) were formed very early on and the rest of the analysed ideas fitted to them quite well. Therefore, it is very probable that also rest of the ideas could fit into these categories. Still, new ideas inside these categories would have probably been found.

7.1.2 Questionnaire

The questionnaire results indicate that users are mostly concerned about their privacy, but do also see the added value that context-awareness brings to messaging. The evaluation also shows that the more actively people send text and multimedia messages the more interested they tend to be in the features provided by context-awareness. Reason for this is most likely that people, who send messages actively, probably have more perspective to see the potential benefits of these scenarios. However, it indicates that it would probably be a good idea to target context-aware communication applications first to these kinds of users.

All things considered, using a web questionnaire as a research method appeared to suit quite well for a quick evaluation of a certain concept. Although it must be said, that a larger number of participants would have given more credibility for the results. Also, one typical problem with questionnaires is that one cannot be

sure whether the respondents have understood the questions and really considered their answers. In this questionnaire, the fact that 25 of all the 48 participants took their time to write meaningful comments about the scenarios shows that at least most of the respondents actually understood the questions and gave their answers with thought.

The participants were given a chance to take part in a lottery of two movie tickets, but only 31 of all the 48 participants took this opportunity. This decision did not seem to have much effect on the quality of the answers. The effort put into answering varied considerably within both the people that participated in the lottery and within the people that did not. It seems that if people find the topic interesting they may participate and share their thoughts broadly even without a reward.

7.1.3 Application Framework

The application framework brought up many issues related to the development of context-aware applications. Most importantly, the choice of which context and content elements to include into the application depends on the purpose of the application. If a general framework is being developed, it should support as many context and content elements as possible. As such, the application framework is not that novel as it only supports basic context elements location and time and a simple text-based search. However, the application framework's general architecture and user interface design support the addition of more context elements.

7.2 Answers to Research Questions

The answers to the research questions are summarised in the following.

7.2.1 How can currently supported context information be used in a textual, remote, and electronic communications scope?

Context information can be used in all phases of the communication. First, it can be used to decide which people to include in the communication and to determine the appropriate device and preferred communication method. The sender may use the recipient's context information to determine whether to approach him/her and the recipient may use the sender's context information to decide whether to accept the communication attempt. In some cases just seeing another person's context change, may result in a communication attempt.

After the communication has been started, context information can be used to determine whether to continue the communication when contexts change and in case of instant communication to determine whether the other person is paying full attention to the communication.

Messages can also be sent so that they are delivered based on contexts, which makes it possible to modify or cancel unsent messages if they become obsolete

before delivery. Reminder messages sent to relevant situations are a special case of messages delivered based on contexts.

Communication can also be completely based on context. Most typical case would be location-based message boards.

7.2.2 What advantages and concerns there are in using currently supported context information in a textual, remote, and electronic communications scope?

The main advantage of using context information is that it makes the communication more efficient and productive. Especially interesting context elements are location, current and future activities, and mood. Context information can also reduce misunderstandings between people from different contexts.

One concern relates to the delivering of messages based on contexts. It cannot be the only option, but urgent messages must be allowed through anyway.

An important question is that since all context data is hard to get automatically will the users have the time and effort to enter required context data manually or to train the system to recognize different contexts. The implemented user interface should be very simple and usable. Also, the automatically collected context data and especially the data manually entered by users may be quite ambiguous.

However, without a doubt the biggest concern is privacy. Users must be able to control and see what context information about them has been sent and to whom. The value that the users gain from revealing the information must also be clearly visible.

7.3. Future Work

The questionnaire results raised some interesting issues about context-aware communication applications. However, questionnaires as such are not sufficient for extracting users' needs and concerns, since it is often relatively easy to say something, but a completely different matter to actually do it. Therefore, these issues should be verified by creating more prototypes and evaluating them with users.

The already created framework for mobile context-based messaging applications should be validated by creating example applications and evaluating them with users. The creation of different applications would also stress the question of how much effort does the framework actually save from the application developers.

Possible scientific forums for publishing the future results on this topic include the following conferences and journals.

Conferences:

- Computer/Human Interaction (CHI) (http://www.chi2008.org/)
- The Ninth Workshop on Mobile Computing Systems and Applications (HotMobile 2008) (http://prisms.cs.umass.edu/hotmobile2008/)
- The 10th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI 2008) (http://mobilehci2008.telin.nl/)
- The Sixth International Conference on Mobile Systems, Applications, and Services (MobiSys 2008) (http://www.sigmobile.org/mobisys/2008/)
- The 9th International Conference on Ubiquitous Computing (UbiComp 2007) (http://www.ubicomp2007.org/)
- 3rd IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2007) (http://www.gel.usherbrooke.ca/WiMob2007/)
- The 5th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MOBIQUITOUS 2008) (http://www.mobiquitous.org/)

Journals:

- ACM Transactions on Computer-Human Interaction
- IEEE Pervasive Computing
- IEEE Wireless Communications
- Journal of Ubiquitous Computing and Intelligence
- Personal and Ubiquitous Computing
- Ubiquitous Computing And Communication Journal

The question that especially deserves further studying is could other context information than location, time, activity, and identities of nearby people be utilized in mobile communications scope and what kind of new possibilities and concerns that would create.

References

Abowd, G., Atkeson, C., Hong, J., Long, S., Kooper, R. and Pinkerton, M., 1997. Cyberguide: A mobile context-aware tour guide. Wireless Networks, 3(5): 421–433.

Alahuhta, P. and Abrahamsson, P., 2006. The Idea Movement: 35 000 Ideas for New Mobile Services for Open Use. http://www.idealiike.fi/en/index.htm.

Asthana, A., Cravatts, M. and Krzyzanowski, P., 1994. An indoor wireless system for personalized shopping assistance. Proceedings of IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, USA, 69-74.

Baldauf, M., Dustdar, S. and Rosenberg, F., 2007. A Survey on Context-Aware Systems. International Journal of Ad Hoc and Ubiquitous Computing, 2(4): 263-277.

Bardram, J.E. and Hansen, T.R., 2004. The AWARE Architecture: Supporting ContextMediated Social Awareness in Mobile Cooperation. Proceedings of the 2004 ACM conference on Computer supported cooperative work, Chicago, Illinois, USA,192–201.

Barkhuus, L. and Dey, A.K., 2003. Location-based services for mobile telephony: a study of users' privacy concerns.

Beresford, A.R. and Stajano, F., 2003. Location Privacy in Pervasive Computing. IEEE Pervasive Computing, 2(1): 46–55.

Brown, P.J., Bovey, J.D. and Chen, X., 1997. Context-aware applications: from the laboratory to the marketplace. IEEE Personal Communications, 4(5): 58-64.

Brown, P.J. and Jones, G.J.F., 2002. Exploiting Contextual Change in Context-Aware Retrieval. Proceedings of the 2002 ACM symposium on Applied computing, Madrid, Spain, 650–656.

Burrell, J. and Gay, G.K., 2002. E-Graffiti: Evaluating Real-World Use of a Context- Aware System. Interacting with Computers, 14: 301–312.

Chen, G. and Kotz, D., 2000. A Survey of Context-Aware Mobile Computing Research. Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College.

Cheverst, K., Mitchell, K., Davies, N. and Smith, G., 2000. Exploiting Context to Support Social Awareness and Social Navigation. ACM SIGGROUP Bulletin, 21(3): 43–48.

Dabbish, L. and Kraut, R.E., 2004. Controlling Interruptions: Awareness Displays and Social Motivation for Coordination. Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work, Chicago, Illinois, USA, 182-191.

Danninger, M., Flaherty, G., Bernardin, K., Ekenel, H.K., Köhler, T., Malkin, R., Stiefelhagen, R. and Waibel, A., 2005. The connector: facilitating context-aware communication. Proceedings of the 7th international conference on Multimodal interfaces, Torento, Italy, 69–75.

Danninger, M., Kluge, T. and Stiefelhagen, R., 2006. MyConnector: Analysis of Context Cues to Predict Human Availability for Communication. Proceedings of the 8th International Conference on Multimodal Interfaces, Banff, Alberta, Canada, 12-19.

Davies, N., Cheverst, K., Mitchell, K. and Friday, A., 1999. Caches in the air: Disseminating tourist information in the GUIDE system. Proceedings of Second IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, USA, 11-19.

Davis, S. and Gutwin, C., 2005. Using Relationship to Control Disclosure in Awareness Servers. Proceedings of Graphics Interface 2005, Victoria, British Columbia, Canada, 145-152.

De Guzman, E.S., Sharmin, M. and Bailey, B.P., 2007. Should I Call Now? Understanding What Context is Considered When Deciding Whether to Initiate Remote Communication via Mobile Devices. Proceedings of the Graphics Interface Conference (GI 2007), Montreal, Canada, 143-150.

De Guzman, E.S., Yau, M., Gagliano, A., Park, A. and Dey, A.K., 2004. Exploring the Design and Use of Peripheral Displays of Awareness Information. Proceedings of the conference on Human Factors and Computing Systems (CHI ’04), Vienna, Austria, 1247-1250.

Dey, A.K. and Abowd, G.D., 2000. CybreMinder: A Context-Aware System for Supporting Reminders. Proceedings of the 2nd International Symposium on Handheld and Ubiquitous Computing (HUC2K), Bristol, UK, 172–186.

Dey, A.K. and Abowd, G.D., 2000. Towards a Better Understanding of Context and Context-Awareness. Proceedings of the Workshop on The What, Who, Where, When, Why and How of Context-Awareness, as part of the Conference on Human Factors in Computing Systems (CHI 2000), The Hague, The Netherlands.

Dey, A.K, Futakawa, M., Salber, D. and Abowd, G.D., 1999. The Conference Assistant: Combining Context-Awareness with Wearable Computing. Proceedings of the 3rd International Symposium on Wearable Computers (ISWC '99), 21-28.

Dey, A.K, Hamid, R., Beckmann, C., Li, I. and Hsu, D., 2004. a CAPpella: Programming by demonstration of context-aware applications. In ACM Conference on Human Factors in Computing Systems (CHI '04), Vienna, Austria.

Dey, A.K. and Mankoff, J., 2005. Designing Mediation for Context-Aware Applications. ACM Transactions on Computer-Human Interaction, 12(1): 53-80.

Floreen, P., Przybilski, M., Nurmi, P., Koolwaaij, J., Tarlano, A., Wagner, M., Luther, M., Bataille, F., Boussard, M., Mrohs, B. and Lau, S., 2005. Towards a Context Management Framework for MobiLife. Proceedings of the 14th IST Mobile & Wireless Communications Summit, Dresden, Germany.

Fogarty, J., Hudson, S.E., Atkeson, C.G., Avrahami, D., Forlizzi, J., Kiesler, S., Lee, J.C. and Yang, J., 2005. Predicting Human Interruptibility with Sensors. ACM Transactions on Computer-Human Interaction, 12(1): 119 - 146.

Fogarty, J., Lai, J. and Christensen, J., 2004. Presence versus Availability: The Design and Evaluation of a Context-Aware Communication Client. International Journal of Human-Computer Studies, 61(3): 299–317.

Frazer, B., Richardson, T. and Harter, A., 1994. Teleporting - making applications mobile. Proceedings of IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, USA, 82-84.

Goertz, M., Ackermann, R., Schmitt, J. and Steinmetz, R., 2004. Context-aware Communication Services: A Framework for Building Enhanced IP Telephony Services. Proceedings of the International Conference on Computer Communications and Networks (ICCCN 2004), Chicago, IL, USA, 535–540.

Google, 2007. Reach out and message someone. Available at http://googleblog.blogspot.com/2007/10/reach-out-and-message-someone.html (referred 16.10.2007)

Gross, T., Braun, S. and Krause, S., 2006. MatchBase: A Development Suite for Efficient Context-Aware Communication. Proceedings of the 14th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP 2006), Montbéliard-Sochaux, France, 308–315.

Göker, A. and Myrhaug, H.I., 2002. User Context and Personalisation. In Workshop Proceedings for the 6th European Conference on Case Based Reasoning.

Hiltunen, K-M., Häkkilä, J. and Tuomela, U., 2005. Subjective Understanding of Context Attributes: A Case Study. Proceedings of the 19th conference of the computer-human interaction special interest group (CHISIG), Canberra, Australia, 1-4.

Hong, J.I. and Landay, J.A., 2004. An Architecture for Privacy-Sensitive Ubiquitous Computing. Proceedings of the 2nd international conference on Mobile systems, applications, and services, Boston, MA, USA, 177–189.

Hull, R., Kumar, B., Lieuwen, D., Patel-Schneider, P.F., Sahuguet, A., Varadarajan, S. and Vyas, A., 2004. Enabling Context-Aware and Privacy-Conscious User Data Sharing. Proceedings of the IEEE International Conference on Mobile Data Management (MDM 2004), Berkeley, CA, USA, 187–198.

IBM, 1999. Pervasive Computing. Systems Journal, 38(4).

Intille, S.S., Bao, L., Tapia, E.M. and Rondoni, J., 2004. Acquiring In Situ Training Data for Context-Aware Ubiquitous Computing Applications. Proceedings of the SIGCHI conference on Human factors in computing systems, Vienna, Austria, 1–8.

Johnson, S., 2007. A Framework for Mobile Context-Aware Applications. BT Technology Journal, 25(2): 106-111.

Jung, Y., Persson, P. and Blom, J., 2005. DeDe: Design and Evaluation of a Context-Enhanced Mobile Messaging System. Proceedings of the SIGCHI conference on Human factors in computing systems, Portland, Oregon, USA, 351–360.

Khalil, A. and Connelly, K., 2005. Improving Cell Phone Awareness by Using Calendar Information. Lecture notes of INTERACT 2005, Rome, Italy, 3585: 588–600.

Khalil, A. and Connelly, K., 2006. Context-aware Telephony: Privacy Preferences and Sharing Patterns. Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work, Banff, Alberta, Canada, 469-478

Kolari, J., Laakko, T., Hiltunen, T., Ikonen, V., Kulju, M., Suihkonen, R., Toivonen, S. and Virtanen, T., 2004. Context-Aware Services for Mobile Users - Technology and User Experiences. Otamedia Oy.

Koolwaij, J., Tarlano, A., Luther, M., Nurmi, P., Mrohs, B., Battestini, A. and Vaidya, R., 2006. Context Watcher - Sharing Context Information in Everyday Life. Proceedings of the IASTED International Conference on Web Technologies, Applications and Services (WTAS 2006), Calgary, Alberta, Canada.

Korpipää, P. and Mäntyjärvi, J., 2003. An Ontology for Mobile Device Sensor-Based Context Awareness. Proceedings of the Fourth International and Interdisciplinary Conference on Modeling and Using Context, Stanford, California, USA, 451-458.

Korpipää, P., Mäntyjärvi, J., Kela, J., Keränen, H. and Malm, E.J., 2003. Managing Context Information in Mobile Devices. Pervasive Computing 2(3): 42-51.

Law, C., Zhang, X., Chan, S. and Wang, C., 2006. Smart Instant Messenger in Pervasive Computing Environments. Proceedings of the First International Conference on Grid and Pervasive Computing (GPC 2006), Taichung, Taiwan, 32-41.

Lederer, S., Beckmann, C., Mankoff, J. and Dey, A.K., 2003. Managing personal information disclosure in ubiquitous computing environments. Technical Report UCB-CSD-03-1257, Computer Science Division, University of California, Berkeley, USA.

Lederer, S., Mankoff, J. and Dey, A.K., 2003. Who Wants to Know What When? Privacy Preference Determinants in Ubiquitous Computing. Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2003), Fort Lauderdale, Florida, USA, 724–725.

Lehikoinen, J.T. and Kaikkonen, A., 2006. PePe Field Study: Constructing Meanings for Locations in the Context of Mobile Presence. Proceedings of the 8th conference on Human-computer interaction with mobile devices and services, Helsinki, Finland, 53–60.

Lei, H., Sow, D. M., Davis, J. S., Banavar, G. and Ebling, M. R., 2002. The Design and Applications of a Context Service. ACM SIGMOBILE Mobile Computing and Communications Review, 6(4): 45–55.

Lei, H. and Ranganathan, A., 2004. Context-aware unified communication. Proceedings of 2004 IEEE International Conference on Mobile Data Management, Berkeley, California, USA, 176–186.

Ludford, P. J., 2006. Sharing Everyday Places I Go While Preserving Privacy. Extended abstracts of Human Factors in Computing Systems (CHI '06), Montreal, Quebec, Canada, 1771-1774.

Ludford, P. J., Frankowski, D., Reily, K., Wilms, K. and Terveen, L., 2006. Because I Carry My Cell Phone Anyway: Functional Location-Based Reminder Applications. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Montreal, Quebec, Canada, 889-898.

Malkin, R., Chen, D., Yang J. and Waibel, A., 2006. Multimodal Estimation of User Interruptibility for Smart Mobile Telephones. Proceedings of the 8th International Conference on Multimodal Interfaces, Banff, Alberta, Canada, 118-125.

Marmasse, N., 1999. ComMotion: A Context-aware Communication System. Proceedings of the Conference on Human Factors in Computing Systems (CHI 99), Pittsburgh, Pennsylvania, USA, 320–321.

Mayrhofer, R., 2004. An Architecture for Context Prediction. PhD Thesis. Johannes Kepler University of Linz, Austria.

Melinger, D., Bonna, K., Sharon, M. and SantRam, M., 2004. Socialight: A Mobile Social Networking System. Interactive poster of the Sixth International Conference on Ubiquitous Computing (UbiComp 2004), Nottingham, England.

Meyer, S. and Rakotonirainy, A., 2003. A Survey of Research on Context-Aware Homes. Conferences in Research and Practice in Information Technology Series, 34(1): 159–168.

Milewski, A.E. and Smith, T.M., 2000. Providing Presence Cues to Telephone Users. Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work, Philadelphia, Pennsylvania, USA, 89-96.

Miller, C.A., Wu, P. and Chapman, M., 2004. The Role of "Etiquette" in an Automated Medication Reminder. Dialogue Systems for Health Communication, 88–96.

Munoz, M.A., Rodriguez, M., Favela, J., Martinez-Garcia, A.I. and Gonzalez, V.M., 2003. Context-Aware Mobile Communication in Hospitals. IEEE Computer Society, 36(9): 38–46.

Myles, G., Friday, A. and Davies, N., 2003. Preserving Privacy in Environments with Location-Based Applications. Pervasive Computing, 2(1): 56–64.

Nagel, K., Kidd, C.D., O'Connell, T., Dey, A.K. and Abowd, G.D., 2001. The Family Intercom: Developing a Context-Aware Audio Communication System. Proceedings of the 3rd International Conference on Ubiquitous Computing (UbiComp 2001), Atlanta, Georgia, USA, 176-183.

Nakanishi, Y., Takahashi, K., Tsuji, T. and Hakozaki, K., 2004. Mobile Communications - iCAMS: A Mobile Communication Tool Using Location and Schedule Information. Pervasive Computing, 3(1): 82-88.

Nakanishi, Y., Tsuji, T., Ohyama, M. and Hakozaki, K., 2000. Context Aware Messaging Service: A Dynamical Messaging Delivery using Location Information and Schedule Information. Personal and Ubiquitous Computing, 4(4): 221–224.

Nikkei Electronics Asia, 2007. Global Mobile Phone Subscribers to Reach 3.3 Billion in 2011. Available at http://techon.nikkeibp.co.jp/article/HONSHI/20070215/127796/. (referred 4.10.2007)

Oulasvirta, A., Raento, M. and Tiitta, S., 2005. ContextContacts: Re-Designing SmartPhone's Contact Book to Support Mobile Awareness and Collaboration. Proceedings of the 7th International Conference on Human

Computer Interaction with Mobile Devices & Services, Salzburg, Austria, 167-174.

Paciga, M. and Lutfiyya, H., 2005. Herecast: An Open Infrastructure for Location-Based Services Using WiFi. IEEE International Conference on Wireless And Mobile Computing, Networking And Communications (WiMob'2005), Montreal, Canada, 4(1): 21–28.

Patil, S. and Lai, J., 2005. Who Gets to Know What When: Configuring Privacy Permissions in an Awareness Application. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Portland, Oregon, USA, 101-110.

Pentland, A., 2004. Healthwear: Medical Technology Becomes Wearable. Computer, 37(5): 42–49.

Raento, M., 2007. Exploring Privacy for Ubiquitous Computing: Tools, Methods and Experiments. PhD Thesis. University of Helsinki, Finland.

Raento, M. and Oulasvirta, A., 2005. Privacy management for social awareness applications. Proceedings of the workshop on Context Awareness for Proactive Systems (CAPS 2005), Helsinki, Finland, 105–114.

Raento, M., Oulasvirta, A., Petit, R. and Toivonen, H., 2005. ContextPhone: A Prototyping Platform for Context-Aware Mobile Applications. Pervasive Computing, 4(2): 51–59.

Ranganathan, A., Campbell, R.H., Ravi, A. and Mahajan, A., 2002. ConChat: A Context-Aware Chat Program. IEEE Pervasive Computing, 1(3): 51–57.

Ranganathan, A. and Lei, H., 2003. Context-Aware Communication. IEEE Computer Society, 36(4): 90–92.

Rantanen, M., Oulasvirta, A., Blom, J., Tiitta S. and Mäntylä, M., 2004. InfoRadar: Group and Public Messaging in the Mobile Context. Proceedings of the third Nordic conference on Human-computer interaction (NordiCHI 2004), Tampere, Finland, 3(5): 131–140.

Reuters, 2007. Is GPS-enabled social networking mobile's future? Available at http://networks.silicon.com/mobile/0,39024665,39167190,00.htm. (referred 26.9.2007)

Salber, D., Dey, A.K. and Abowd, G.D., 1999. The Context Toolkit: Aiding the Development of Context-Enabled Applications. Proceedings of the 1999 Conference on Human Factors in Computing Systems (CHI '99), Pittsburgh, Pennsylvania, USA, 434–441.

Schilit, B., Hilbert, D.M. and Trevor, J., 2002. Context-aware Communication. IEEE Wireless Communications, 9(5): 46–54.

Schilit, B.N., 1995. A Context-Aware System Architecture for Mobile Distributed Computing. PhD Thesis. Columbia University.

Schilit, B.N., Adams, N. and Want, R., 1994. Context-Aware Computing Applications. Proceedings of IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, California, USA, 85–90.

Schmidt, A., Aidoo, K.A., Takaluoma, A., Tuomela, U., Laerhoven, K.V. and de Velde, W.V., 1999. Advanced interaction in context. Proceedings of First International Symposium on Handheld and Ubiquitous Computing, HUC'99, Karlsruhe, Germany, 89-101.

Schmidt, A., Beigl, M. and Gellersen H-W., 1999. There is more to Context than Location. Computers and Graphics Journal, 23(6): 893–902.

Schmidt, A., Stuhr, T. and Gellersen, H., 2001. Context-Phonebook - Extending Mobile Phone Applications with Context. Third Mobile HCI Workshop, Lille, France.

Schmidt, A., Takaluoma, A. and Mäntyjärvi, J., 2000. Context-Aware Telephony Over WAP. Personal Technologies, 4(4): 225–229.

Strang, T, Linnhoff-Popien, C. and Frank, K., 2003. CoOL: A Context Ontology Language to Enable Contextual Interoperability. Lecture Notes in Computer Science, 28(93): 236-247.

Tamminen, S., Oulasvirta, A., Toiskallio, K. and Kankainen, A., 2004. Understanding mobile contexts. Personal and Ubiquitous Computing, 8(2): 135–143.

Tang, J.C., 2007. Approaching and Leave-Taking: Negotiating Contact in Computer-Mediated Communication. ACM Transactions on Computer-Human Interaction, 14(1), Article No. 5.

Tang, J.C., Yankelovich, N., Begole, J.,Van Kleek, M., Li, F. and Bhalodia, J., 2001. ConNexus to Awarenex: Extending Awareness to Mobile Users. Proceedings of the SIGCHI Conference on Human factors in Computing Systems, Seattle, Washington, USA, 221-228.

Ter Hofte, G.H., Otte, R., Kruse, H.C.J. and Snijders, M., 2004. Context-aware communication with Live Contacts. Supplementary Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW 2004), Chicago, Illinois, USA.

The Free Dictionary, 2007. Available at http://www.thefreedictionary.com/context. (referred 17.9.2007)

van Sinderen, M.J., van Halteren, A.T., Wegdam, M., Meeuwissen, H.B. and Eertink, E.H., 2006. Supporting Context-Aware Mobile Applications: An Infrastructure Approach. IEEE Communications Magazine, 44(9): 96-104.

Wang, X.H., Zhang, D.Q., Gu, T. and Pung, H.K., 2004. Ontology Based Context Modeling and Reasoning Using OWL. Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, Orlando, Florida, USA, 18-22.

Want, R., Hopper, A., Falcao, V. and Gibbons, J., 1992. The Active Badge Location System. ACM Transactions on Information Systems, 10(1): 91–102.

Want, R., Schilit, B.N., Adams, N.I., Gold, R., Petersen, K., Goldberg, D., Ellis, J.R. and Weiser, M., 1995. An overview of the PARCTAB ubiquitous computing experiment. IEEE Personal Communications, 2(6): 28-43.

Weiser, M., 1991. The Computer for the 21st Century. Scientific American.

Weiser, M. and Brown, J.S., 1995. Designing Calm Technology.

Wilson, D.H., Wyatt, D. and Philipose, M., 2005. Using Context History for Data Collection in the Home. Proceedings of the 1st International Workshop on Exploiting Context Histories in Smart Environments (ECHISE 2005), Munich, Germany.

Yan, H. and Selker, T., 2000. Context-aware office assistant. Proceedings of the 2000 International Conference on Intelligent User Interfaces (IUI 2000), New Orleans, LA, USA, 276-279.

Zhou, C., Ludford, P., Frankowski, D. and Terveen, L., 2005. How Do People's Concepts of Place Relate to Physical Locations? Proceedings of the Tenth IFIP TC13 International Conference on Human-Computer Interaction (INTERACT 2005), Rome, Italy, 886-898.

Appendix A: Web Questionnaire

Note, that this is a translated version. The original questionnaire that the users answered was in Finnish.

At first, give some background information.

Age:

- ○ Under 18
- ○ 18-21 years
- ○ 22-25 years
- ○ 26-30 years
- ○ Over 30 years

Gender:

- ○ Male
- ○Female

How often do you send text messages?

- ○ Never
- ○ Rarely
- ○ Once a week
- ○ Several times a week
- ○ Once a day
- ○ Several times a day

How often do you send multimedia messages?

- ○ Never
- ○ Rarely
- ○ Once a week
- ○ Several times a week
- ○ Once a day
- ○ Several times a day

In general, context information means any information that is typical for a certain situation. In this questionnaire context information can be understood to be composed of the following elements:

- Location information (e.g., at home)
- Time (e.g., 08:00-16:00)
- Activity (e.g., having dinner)
- Surrounding persons (e.g., Matt and Mary)

Context information may consist of a single element (e.g., at home) or from a combination of multiple elements (e.g., at home having dinner with Matt and Mary). Context information could be utilized in many ways when communicating with mobile phones, like when sending text or multimedia

messages. In the following several possibilities that relate to the utilization of context information are presented. Using the multiple choices evaluate each possibility based on how useful you find it. If you find the possibility at least somewhat useful, please write to the comments field in which situations for example the possibility would be useful to you. You can also freely write other comments about the possibilities.

1. Seeing other person's current context information
○ No use ○ Little use ○ Some use ○ Much use
Comments:

2. Restricting what context information about you other persons are allowed to see in different situations
○ No use ○ Little use ○ Some use ○ Much use
Comments:

3. Seeing what context information about you has been sent to others and when
○ No use ○ Little use ○ Some use ○ Much use
Comments:

4. Automatic activation of your own context information instead of having to manually change them every time your context changes
○ No use ○ Little use ○ Some use ○ Much use
Comments:

5. Notification as a sound, small icon etc. every time your automatically activated context changes
○ No use ○ Little use ○ Some use ○ Much use
Comments:

6. Notification as a sound, small icon etc. every time a certain person's context changes
○ No use ○ Little use ○ Some use ○ Much use
Comments:

7. Sending of messages so that they are delivered to the recipient only after the sender is in a certain context
○ No use ○ Little use ○ Some use ○ Much use
Comments:

8. Sending of messages so that they are delivered to the recipient only after the recipient is in a certain context
○ No use ○ Little use ○ Some use ○ Much use
Comments:

9. Cancelling or modifying messages before sender and/or recipient have been in such contexts that the messages could have been sent
○ No use ○ Little use ○ Some use ○ Much use
Comments:

10. Automatic sending of messages repeatedly every time the sender and/or recipient arrive at certain situations

○ No use ○ Little use ○ Some use ○ Much use

Comments:

11. Seeing whether the message has already been delivered to the recipient and whether the recipient has already read the message

○ No use ○ Little use ○ Some use ○ Much use

Comments:

12. Sending messages to oneself to certain situations, e.g., as reminders

○ No use ○ Little use ○ Some use ○ Much use

Comments:

13. Leaving messages to certain places for anyone that arrives at the same place to read

○ No use ○ Little use ○ Some use ○ Much use

Comments:

Appendix B: Questionnaire Charts

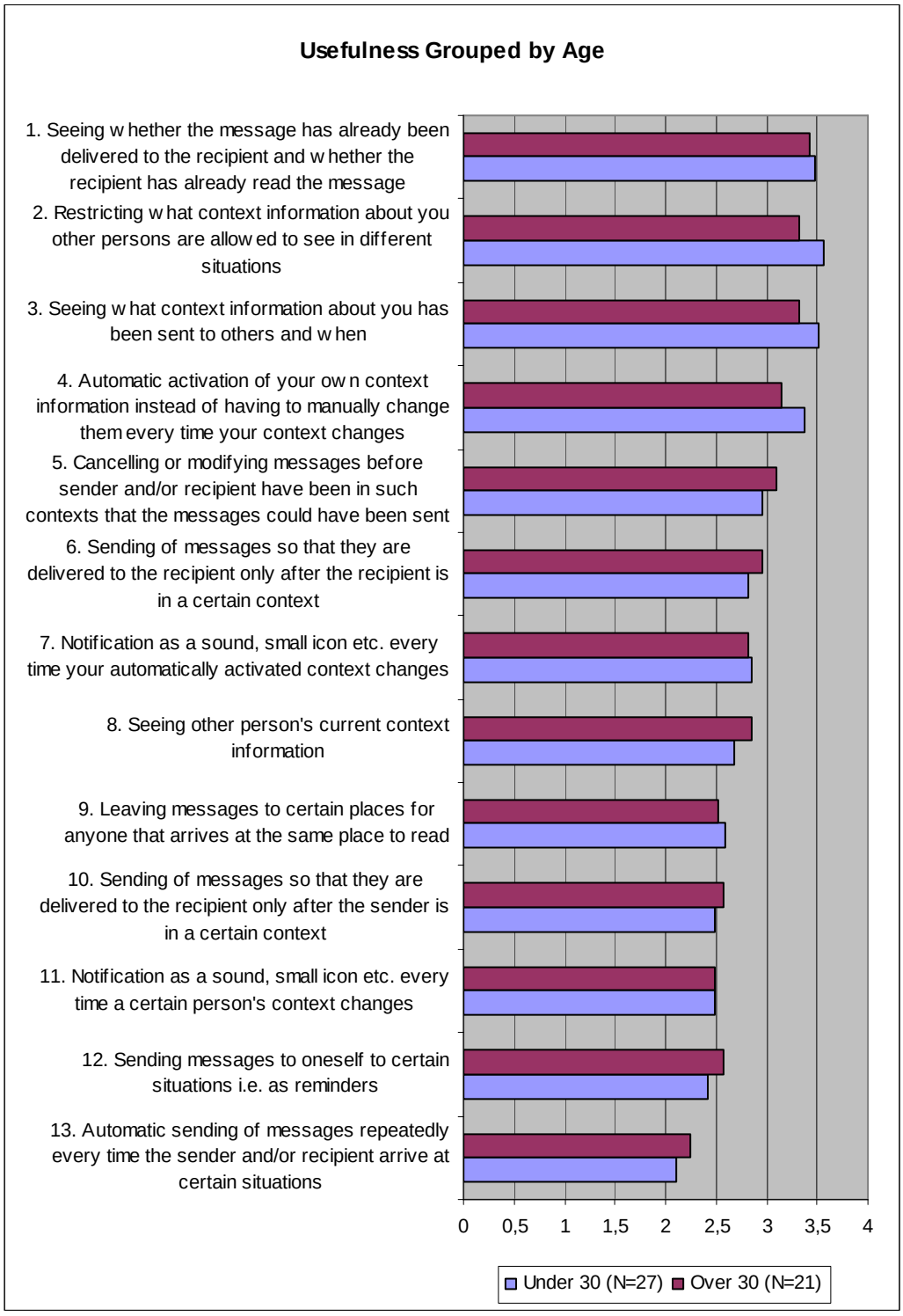

Figure 9. How useful different scenarios are considered by the users grouped by age on a scale of 1–4, where 1 = no use, 2 = little use, 3 = some use and 4 = much use.

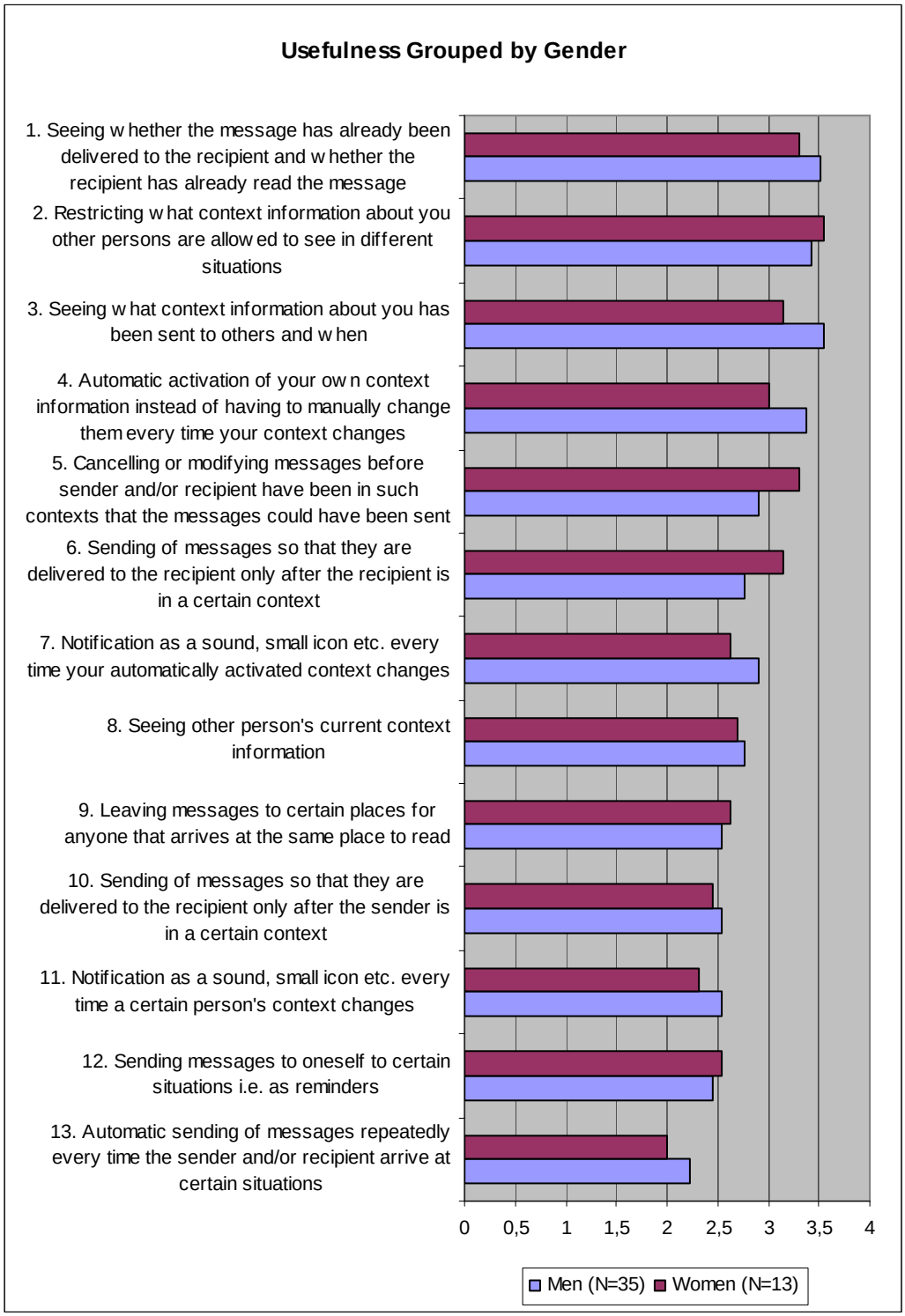

Figure 10. How useful different scenarios are considered by the users grouped by gender on a scale of 1–4, where 1 = no use, 2 = little use, 3 = some use and 4 = much use.

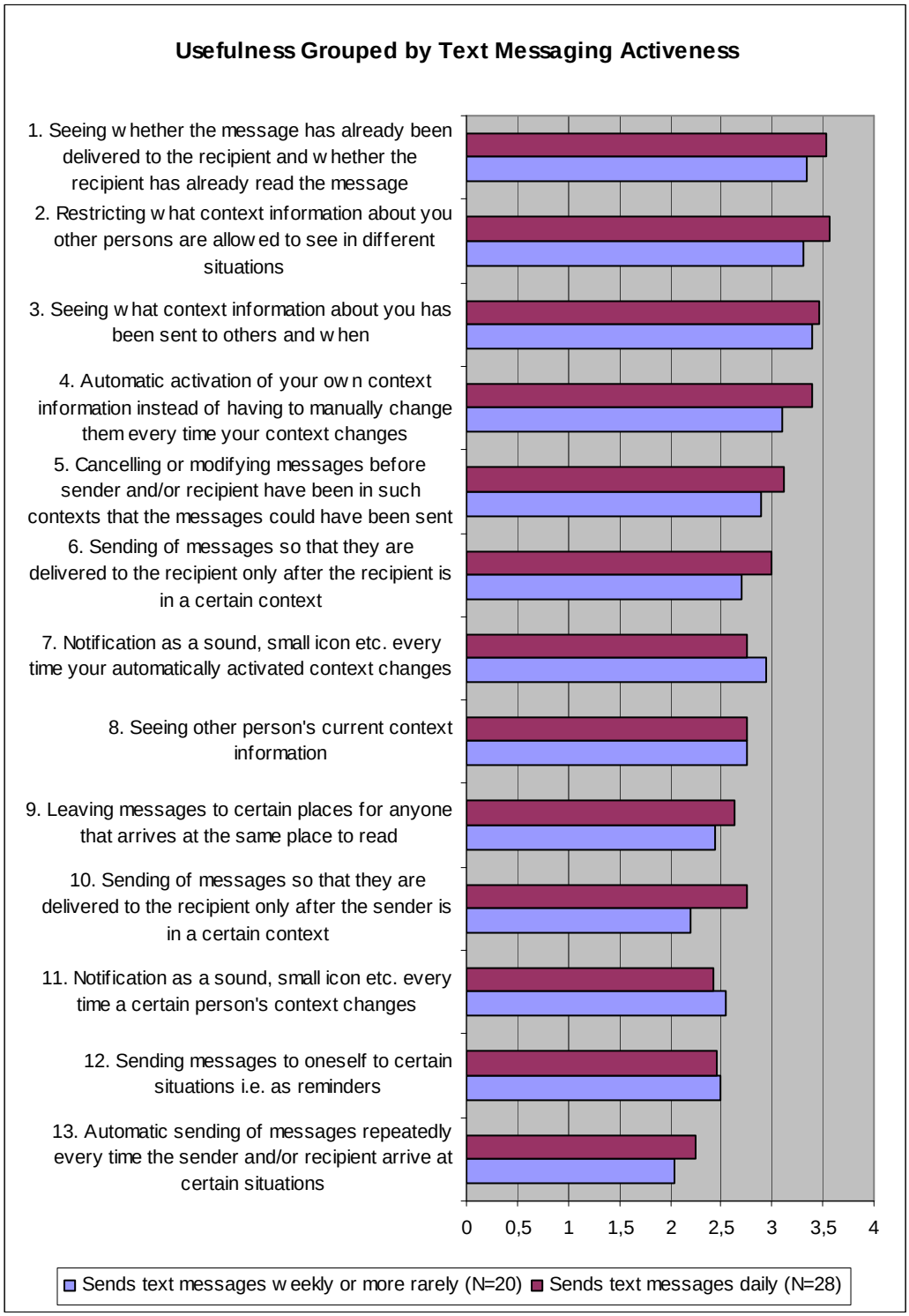

Figure 11. How useful different scenarios are considered by the users grouped by text messaging activeness on a scale of 1–4, where 1 = no use, 2 = little use, 3 = some use and 4 = much use.

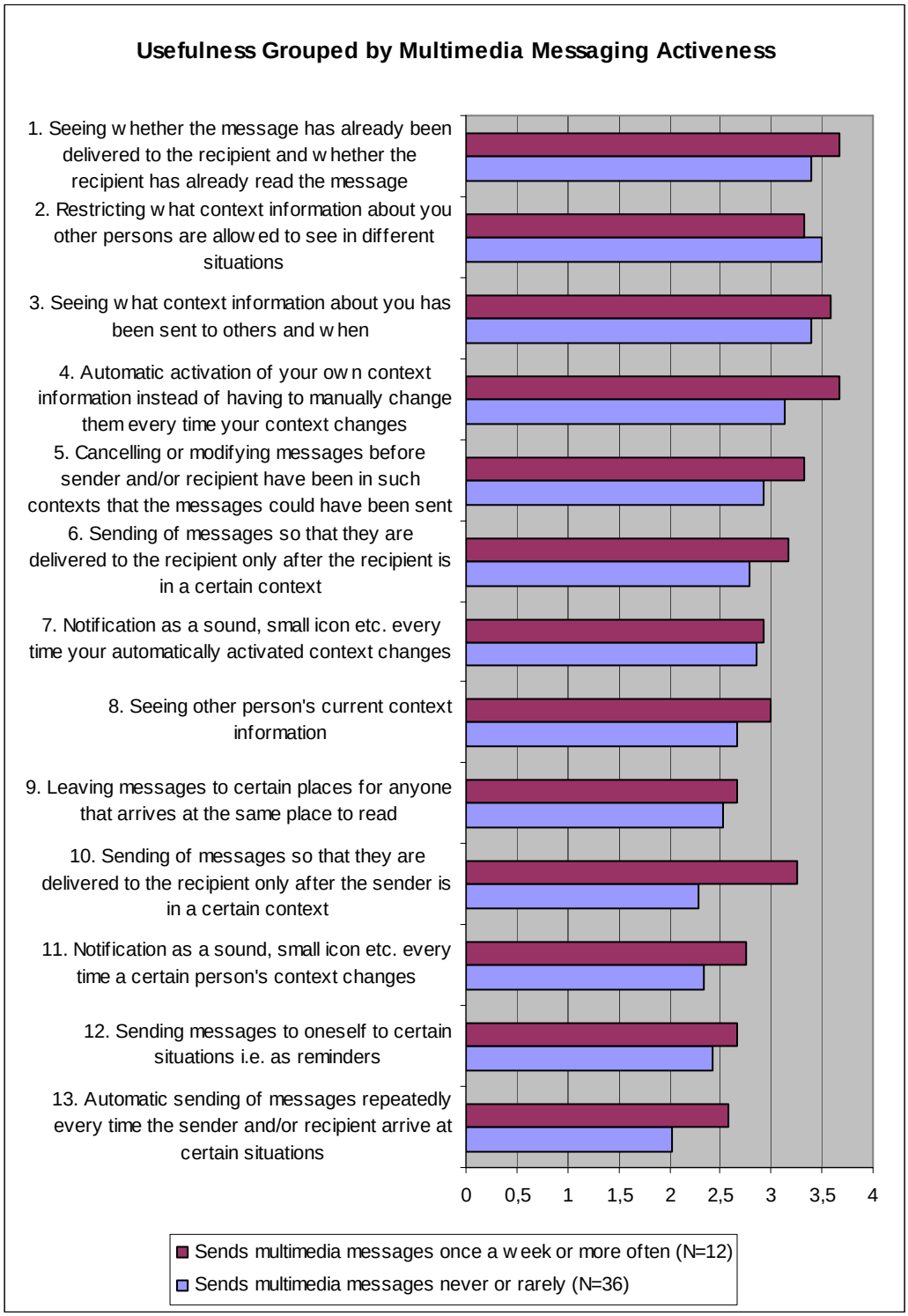

Figure 12. How useful different scenarios are considered by the users grouped by multimedia messaging activeness on a scale of 1–4, where 1 = no use, 2 = little use, 3 = some use and 4 = much use.

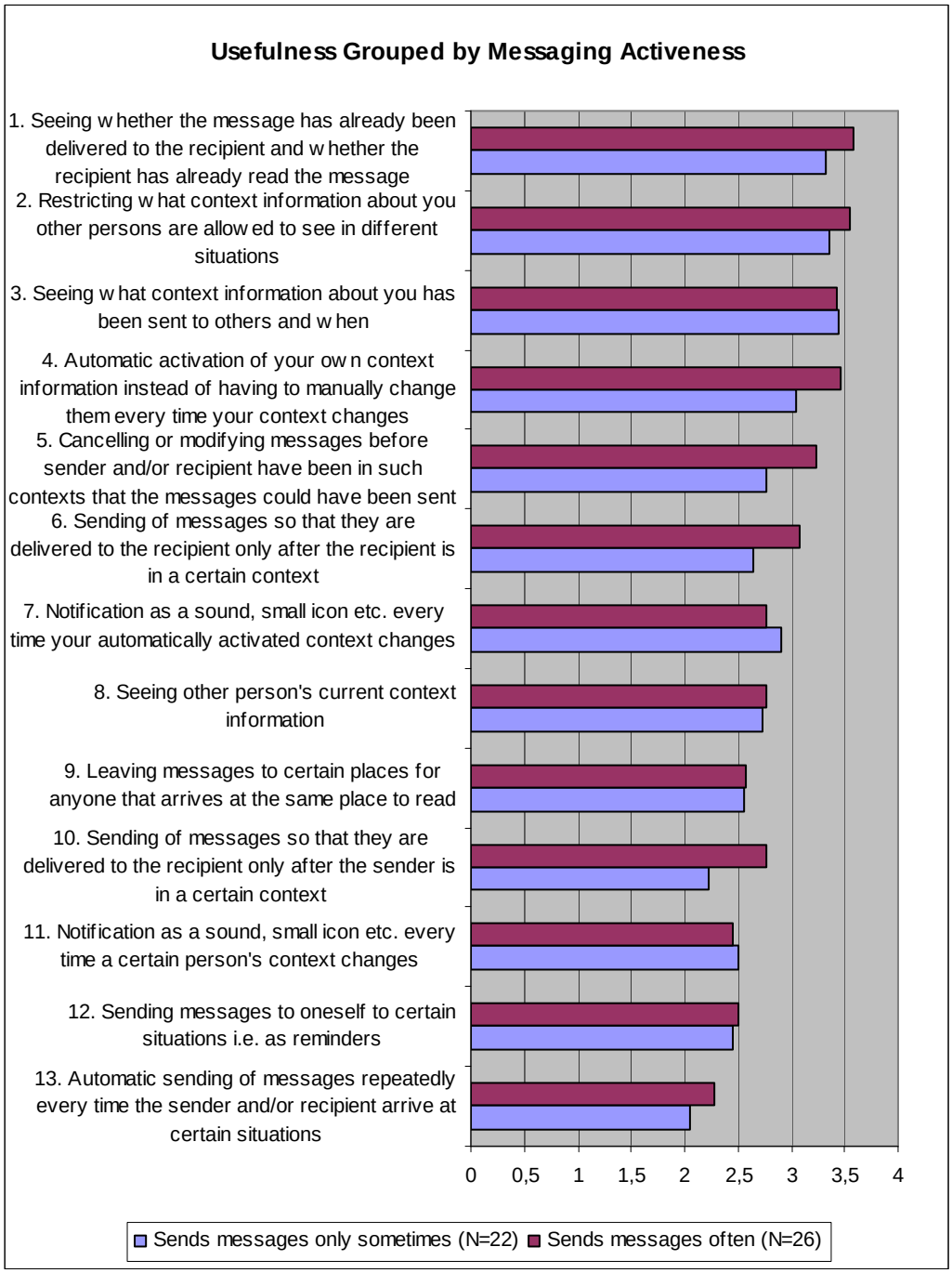

Figure 13. How useful different scenarios are considered by the users grouped by messaging activeness on a scale of 1–4, where 1 = no use, 2 = little use, 3 = some use and 4 = much use.

Appendix C: Example Application

This appendix presents an example application that uses the application framework described in Paper II. The presented application is the same as in the paper and its purpose is to report and discuss problems of public spaces.

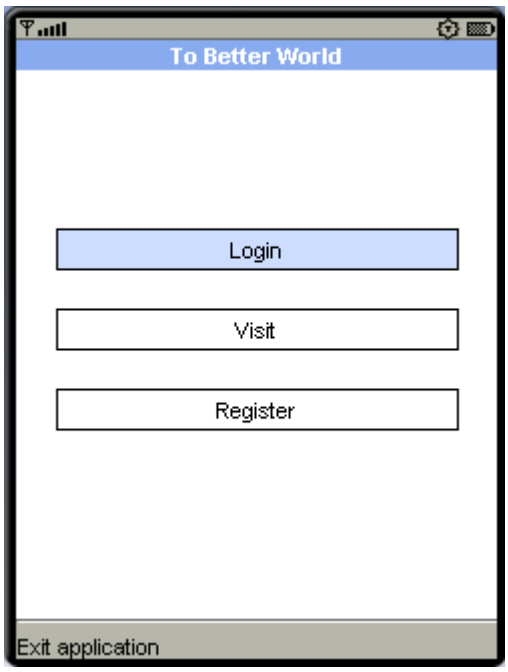

Figure 14. The start page of the application allows the user to login, visit or register to the service. A visitor can browse the messages left to the service, but cannot comment them or send new messages. In the server it can be configured whether visitors are allowed at all. If they are not allowed, this page will only display login and register actions.

Figure 15. In the registration page the necessary information is asked from the user. Which information is asked and which are mandatory can be configured in the server.

Figure 16. After the registration, the user gets to the main page. However, next time the user can login using the login page. If so desired the username and password can be stored to the mobile device's memory so they do not have to be entered every time.

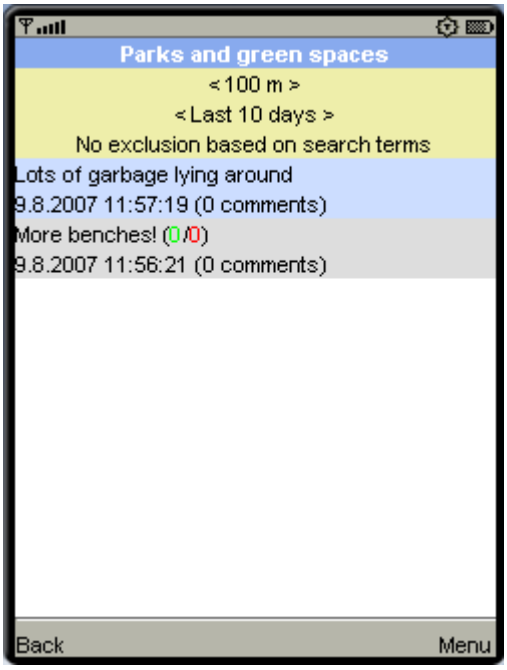

Figure 18. By choosing a certain category, one can see a list of all the messages that it contains. The list shows messages' titles, times when the messages have been sent, and the number of comments left to the messages. Also, if the message has been sent as a poll, one can see the number of people that have voted for and against.

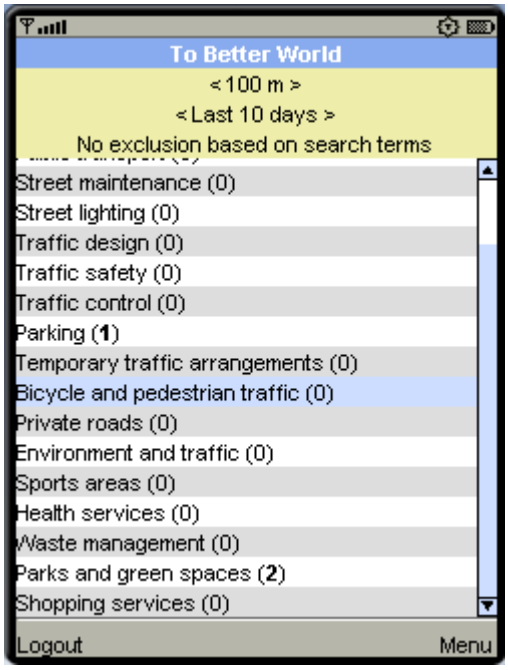

Figure 17. The main page lists different categories to which messages can be sent. The numbers in brackets indicate how many messages there are in each category with the given location, date and search term specifications.

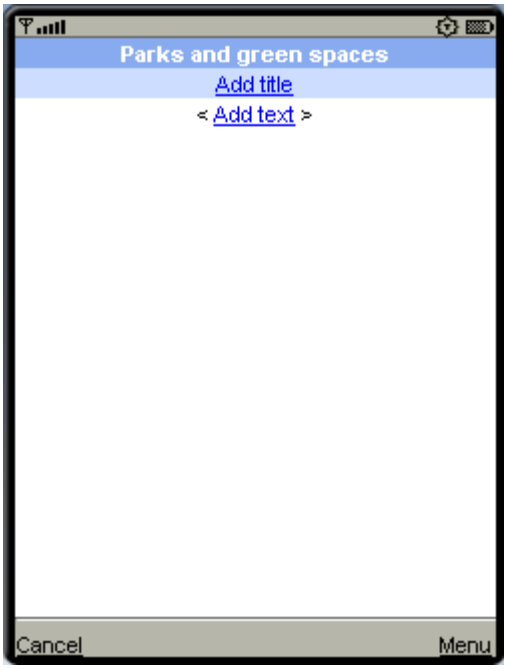

Figure 19. New message is constructed by giving it a title and some content. Possible content elements are texts, images and audios.

Figure 20. An image is added by taking a picture with the mobile device's camera.

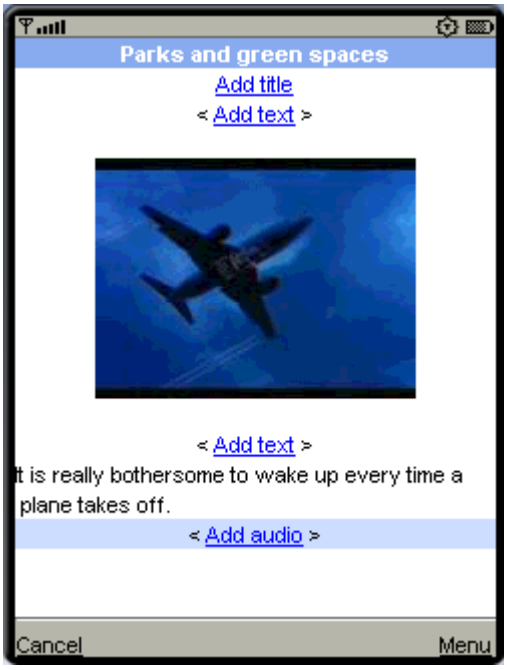

Figure 22. The captured image and added text are shown in the message. They can be removed through the menu if so desired. New content elements can also further be added all around the message.

Figure 21. Similarly, one can add a text element to the message.

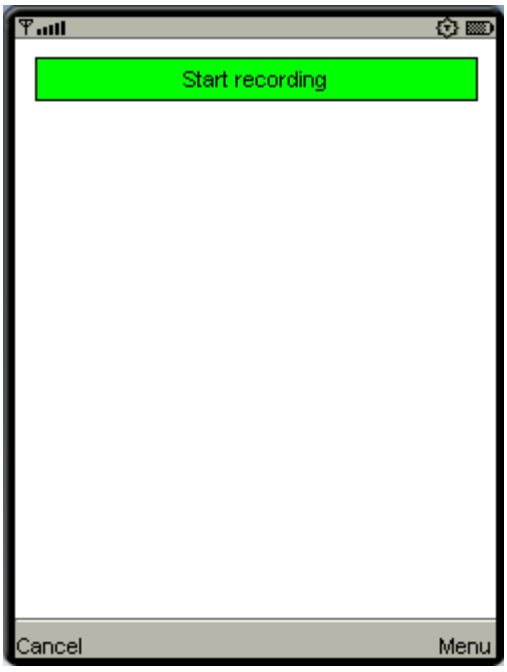

Figure 23. An audio element is added by recording it using the mobile device's microphone.

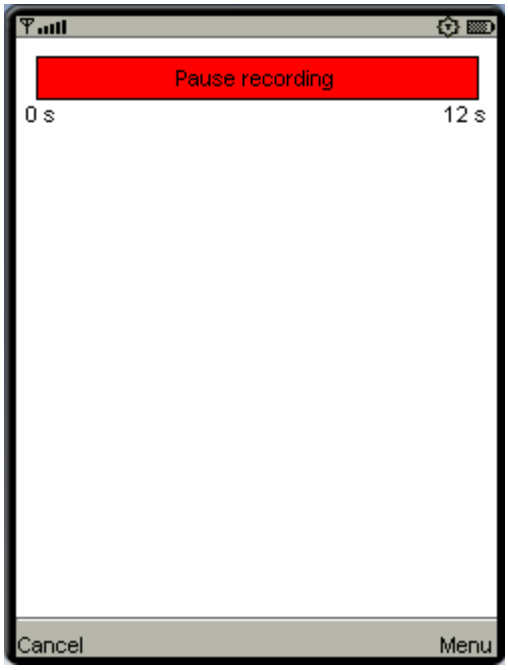

Figure 24. The recording can be paused every now and then and finished by selecting the appropriate command from the menu.

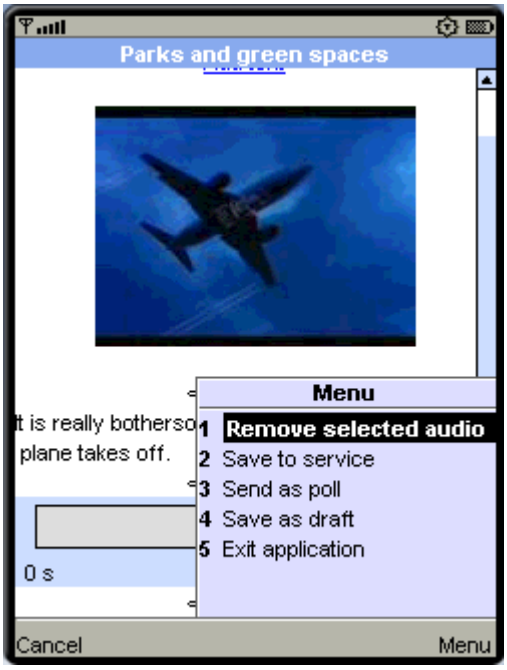

Figure 26. Other menu commands allow saving the message to the service, sending it as a poll, or saving it as a draft. In the server, it can be configured whether polls are allowed. If not, the action to send the message as a poll is not displayed in the menu.

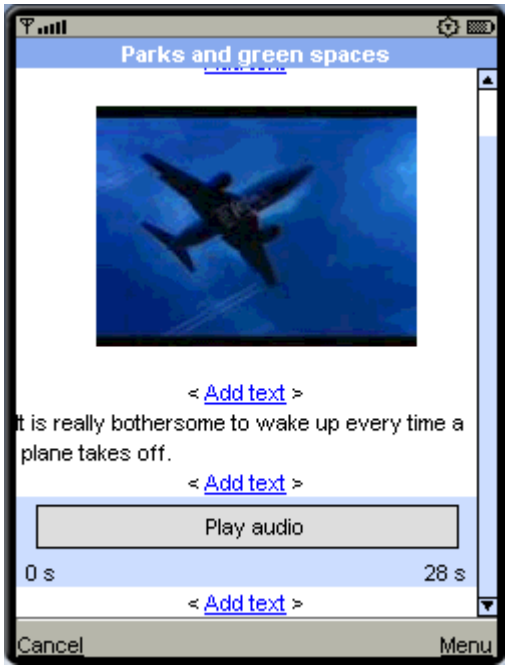

Figure 25. The recorded audio is added to the message where it can be removed through the menu or played back by selecting it..

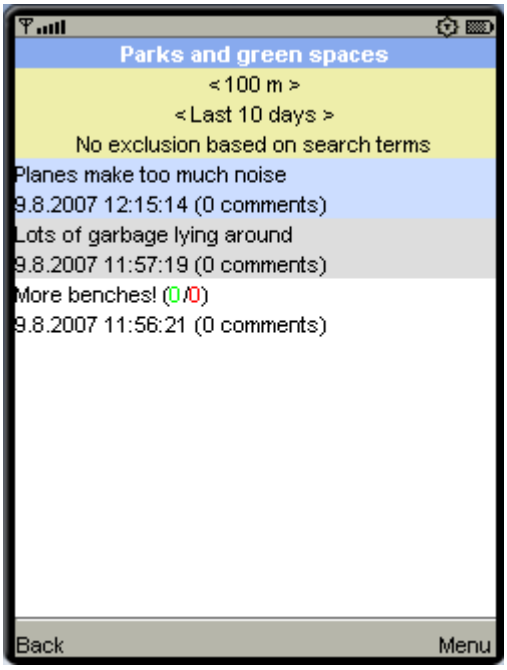

Figure 27. The sent message is displayed in the list. The list is arranged based on time so that the newest message is on top.

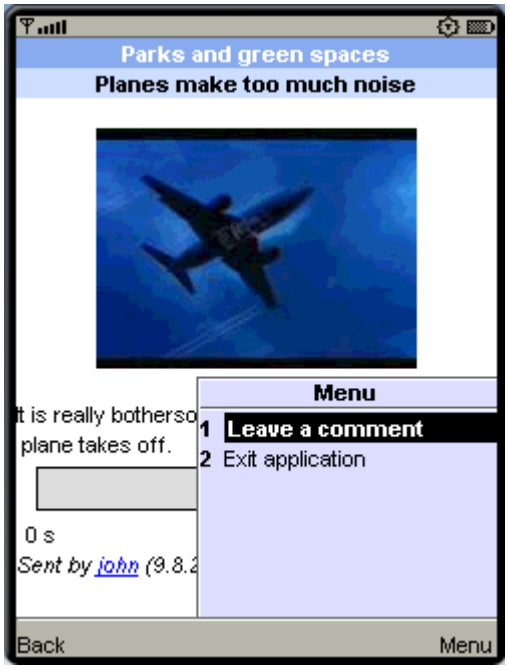

Figure 28. The message can be viewed by selecting it. One can also leave a comment to the message. The message sender's name leads to a page that displays information about the user.

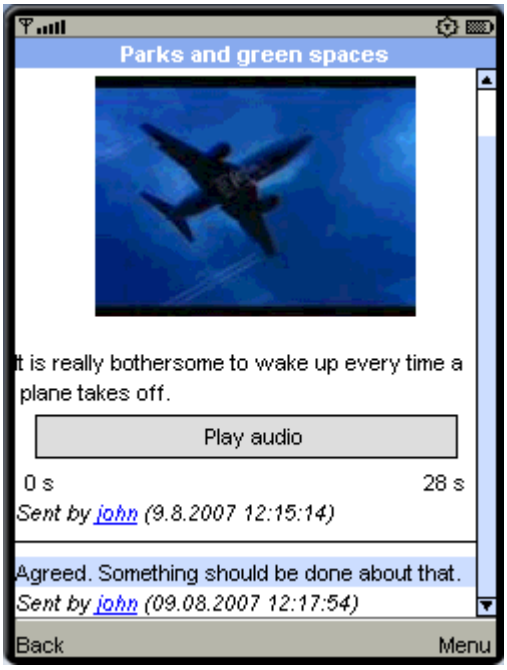

Figure 30. Added comments are shown below the message. Comment senders' names also lead to pages that display information about them.

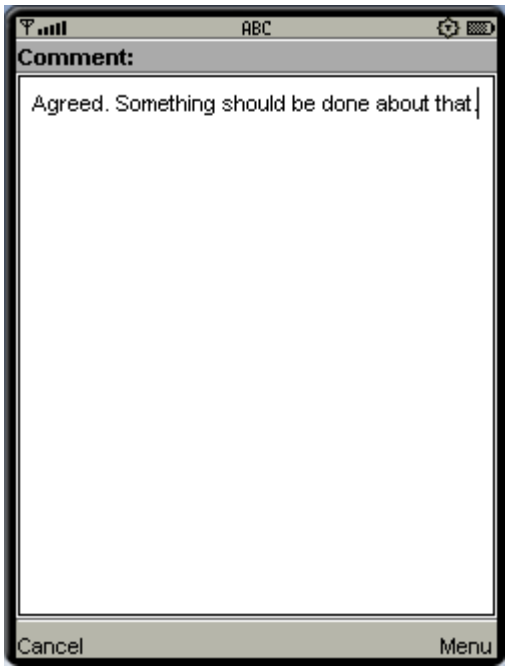

Figure 29. A comment can be added by writing the desired comment as a text.

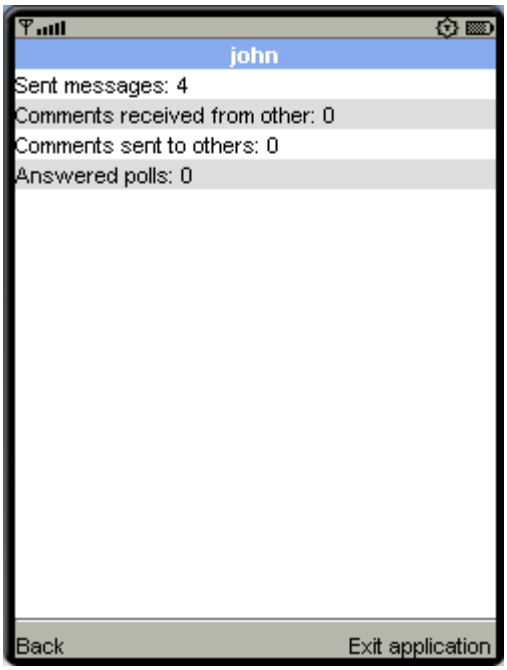

Figure 31. The information shown about the user includes number of sent messages, number of received and sent comments, and number of answered polls. What information is shown can be configured in the server.

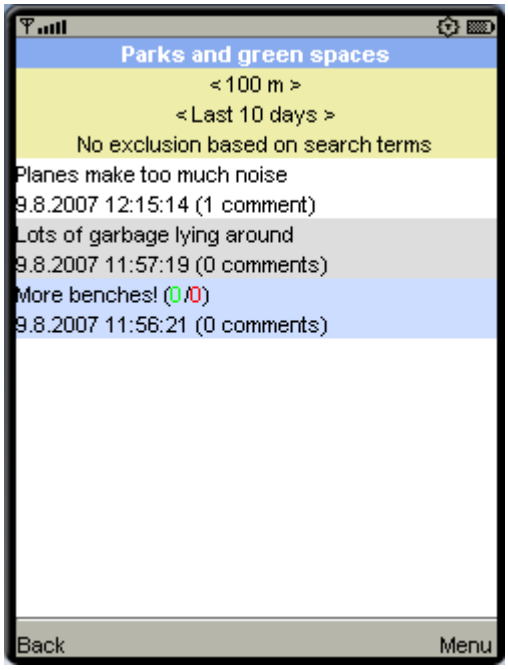

Figure 32. A message sent as a poll can also be chosen from the list.

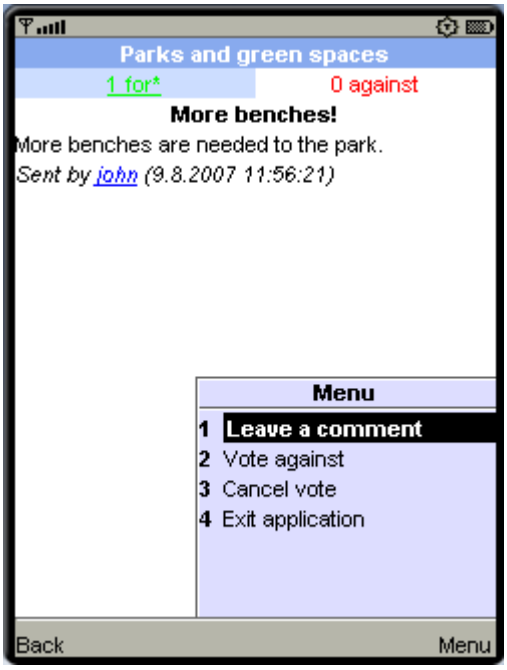

Figure 34. If one votes for, one can see his vote marked with *. The menu then offers a possibility to change one's mind and vote against or cancel the vote. By selecting the for/against field on top of the message, one can see a list of users who have voted like that.

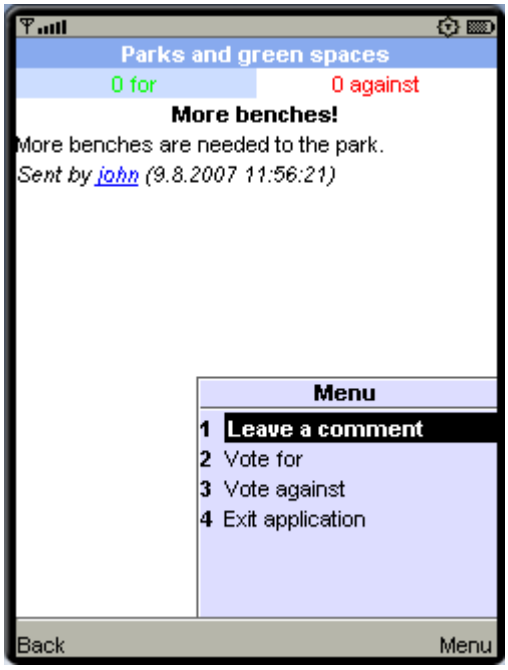

Figure 33. On top of a message sent as a poll are displayed the number of users that have voted for and against the poll. The menu offers the commands for voting.

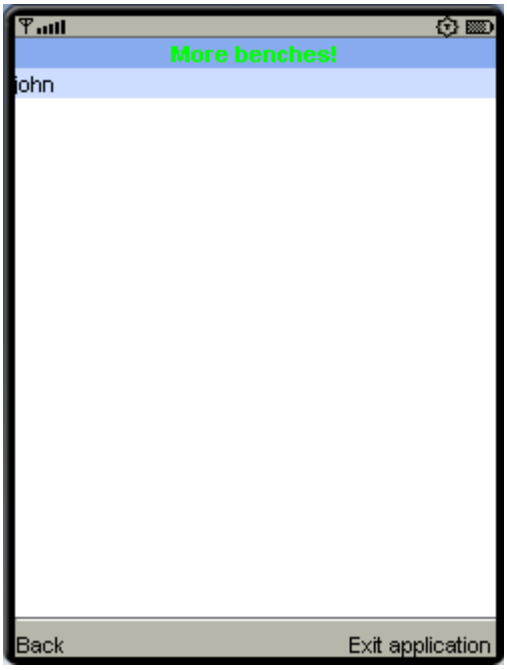

Figure 35. By selecting the for field, a list of users that have voted for is shown. Individual user's information can further be viewed in the same way as in the message and comment senders' case.

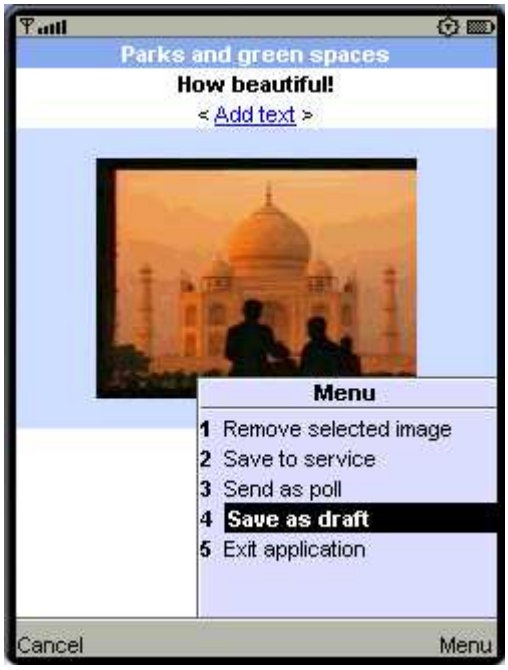

Figure 36. A message can also be saved as draft.

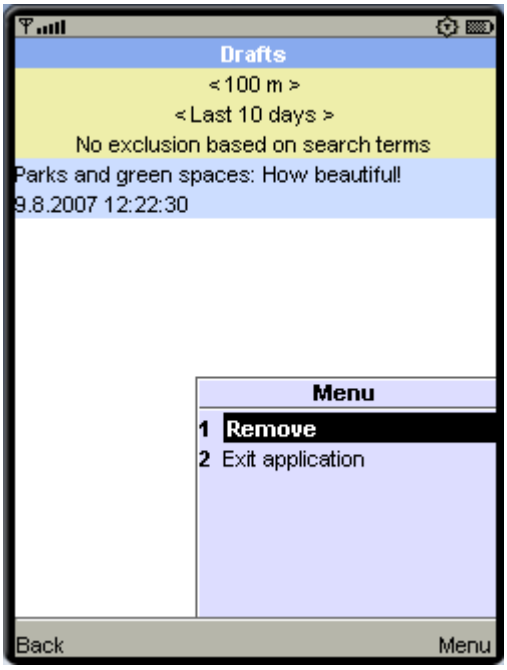

Figure 38. The drafts are shown as a list in the same way as sent messages. The user can remove the draft or modify it by selecting it.

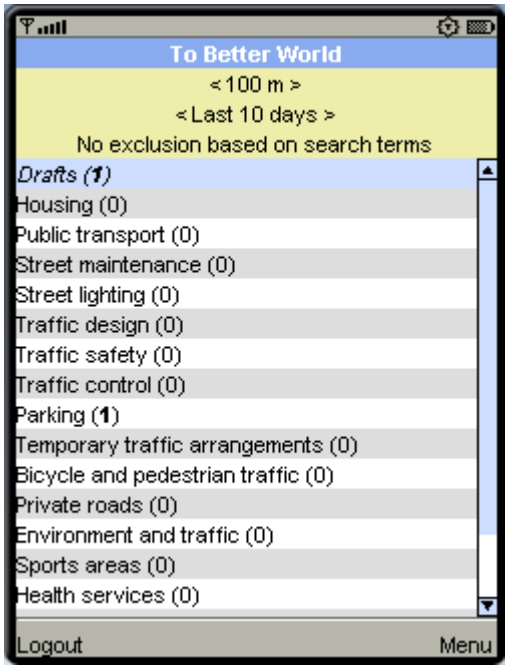

Figure 37. Drafts are shown in the main page same way as other categories. However, only the logged in user sees his/her own drafts. Same location, date and search term specifications can be used to restrict the shown drafts.

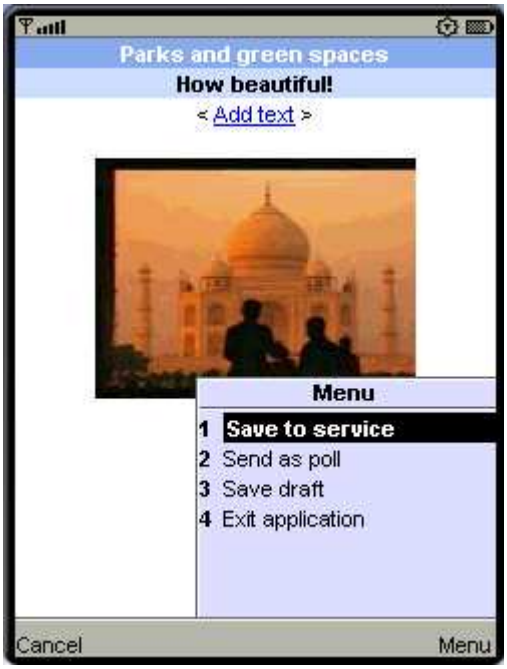

Figure 39. The draft can be modified as regular messages. It can be saved to the service, sent as a poll or further saved as draft.