

**UNIVERSITY OF OSLO**  
**Department of Informatics**

**Mobile Presence**  
**Information Sharing**

Communicating by  
sharing presence  
information

Master thesis

Jens-Martin Grønne

2nd May 2007





---

## Preface

This master thesis is the result of research done in the period from August 2006 to April 2007 at the University of Oslo, Department of Informatics. The area of research is Design of information systems with focus on presence information sharing between mobile users.

I would like to express my gratitude to everybody that has contributed to this work. All comments, constructive critic, support and helpful pointers have improved the quality of this work.

I would especially like to thank Werner Eriksen, my external advisor at Telio Telecom AS, for all support and guidance. The work with this thesis would not have been possible without the help of the very knowledgeable people working in the research and development department of Telio.

I would also like to thank my internal advisor at the Department of Informatics; Jo Herstad for all constructive critic and guidance throughout the work with this thesis.

Jens-Martin Grønne

May 2, 2007

---

---

# Abstract

This is a master thesis written at the University of Oslo, Department of Informatics.

The objective of this thesis is to explore the new open standard framework SIMPLE (Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions) and its ability to provide presence mediated communication between mobile users.

To discuss central concepts like *communication*, *mobility*, *presence*, *context*, *awareness* and *interaction overload* I have done a literature study and I present important theories and models to explain these concepts.

For giving an overview over new communication options with both voice over IP and GSM available on mobile phones I have made several use cases and scenarios describing these new possibilities

To test the abilities of the SIMPLE framework I have made a working prototype setup which allows for sharing of presence information between users. And I also discuss issues with communication that SIMPLE can help resolve.

---

---

# Index

<b>1</b>	<b>INTRODUCTION .....</b>	<b>1</b>
1.1	MOTIVATION .....	1
1.2	PROBLEM AREA.....	3
1.3	PROBLEM DEFINITION .....	3
1.4	OVERVIEW .....	5
<b>2</b>	<b>METHODOLOGY .....</b>	<b>7</b>
2.1	LITERATURE STUDIES.....	7
2.2	USE CASE DRIVEN DEVELOPMENT AND SCENARIOS .....	8
2.3	PROTOTYPING .....	9
<b>3</b>	<b>COMMUNICATION THEORY .....</b>	<b>11</b>
3.1	BACKGROUND INFORMATION.....	11
3.2	COMMUNICATION MODEL .....	11
3.3	MOBILITY .....	15
3.4	PRESENCE .....	16
3.4.1	<i>Physical and social presence.....</i>	<i>17</i>
3.4.2	<i>Measuring presence .....</i>	<i>18</i>
3.5	CONTEXT .....	18
3.5.1	<i>Context sharing .....</i>	<i>21</i>
3.6	AWARENESS .....	23
3.7	INTERACTION OVERLOAD.....	25
3.7.1	<i>Communication Overflow (CO) .....</i>	<i>26</i>
3.7.2	<i>Communication Deficiency (CD) .....</i>	<i>28</i>
<b>4</b>	<b>COMMUNICATION TECHNOLOGY.....</b>	<b>33</b>
4.1	MOBILE PHONES .....	33
4.1.1	<i>History.....</i>	<i>33</i>
4.1.2	<i>Technology.....</i>	<i>34</i>
4.2	VOICE OVER IP .....	35
4.2.1	<i>History.....</i>	<i>35</i>
4.2.2	<i>Technology.....</i>	<i>36</i>
4.3	FIXED TO MOBILE CONVERGENCE / IMS.....	38
4.3.1	<i>Network convergence .....</i>	<i>39</i>
4.3.2	<i>Terminal convergence .....</i>	<i>39</i>
4.3.3	<i>Service convergence.....</i>	<i>39</i>
4.3.4	<i>IMS.....</i>	<i>39</i>

---

4.4	SIMPLE.....	40
4.4.1	Protocol.....	40
4.4.2	PIDF.....	42
4.4.3	RPID.....	44
4.4.4	XCAP.....	45
<b>5</b>	<b>SIMPLE PROTOTYPE .....</b>	<b>47</b>
5.1	USE CASES AND SCENARIOS .....	47
5.1.1	Connectivity based routing.....	47
5.1.2	Least cost routing.....	48
5.1.3	Context / availability routing.....	49
5.1.4	Security based routing.....	50
5.1.5	Selecting a use case.....	51
5.2	MOBILE USER INTERFACE.....	51
5.3	PROTOTYPE SETUP .....	53
5.3.1	Architecture.....	54
5.3.2	Server – Sip Express Router (SER).....	54
5.3.3	Client – X-Lite .....	55
5.3.4	Configuration .....	56
5.3.5	Testing .....	57
<b>6</b>	<b>DISCUSSION.....</b>	<b>59</b>
6.1	PROTOTYPE.....	59
6.1.1	Results .....	60
6.1.2	Presence information sharing.....	61
6.2	VOIP AND GSM ROUTING .....	61
6.3	INTERACTION OVERLOAD.....	63
6.3.1	Communication overflow.....	63
6.3.2	Communication deficiency.....	64
6.4	PRIVACY .....	65
<b>7</b>	<b>CONCLUSION .....</b>	<b>67</b>
7.1	FUTURE WORK .....	68



---

# Figures

FIGURE 1 - PROBLEM AREA .....	3
FIGURE 2 - LASSWELL'S COMMUNICATION MODEL.....	12
FIGURE 3 - SHANNON-WEAVERS' COMMUNICATION MODELL .....	13
FIGURE 4 - CONTEXTUAL COMMUNICATION MODELL.....	14
FIGURE 5 - SKYPE - A CONTEXT-AWARE APPLICATION .....	23
FIGURE 6 - COMMUNICATION OVERFLOW .....	27
FIGURE 7 - COMMUNICATION DEFICIENCY (LJUNGBERG AND SØRENSEN 1998).....	30
FIGURE 8 - OVERVIEW OF SIP NETWORK (CISCO 2005) .....	37
FIGURE 9 - BASIC SIP CALL .....	38
FIGURE 10 - SIMPLE SUBSCRIPTION .....	41
FIGURE 11 - SIMPLE PRESENCE UPDATE.....	42
FIGURE 12 - XCAP AUTHENTICATION.....	46
FIGURE 13 - PHONEBOOK WITH PRESENCE .....	52
FIGURE 14 - PHONEBOOK WITH LIMITED COMMUNICATION METHODS .....	53
FIGURE 15 - SIMPLE ARCHITECTURE .....	54
FIGURE 16 - PROTOTYPE SERVER PLATFORM .....	55
FIGURE 17 - X-LITE CLIENT.....	56

---

# 1 Introduction

## *1.1 Motivation*

Today's information technology gives us the opportunity to communicate with co-workers, friends and family wherever we may be. The development of communication technologies have led to the point where we are no longer bound to a certain place or area to be able to communicate with others. These developments in communication technology enable us to communicate with anyone, anywhere at any time.

The demand for always being available has led to a tremendous development in mobile communication technology. Only a few year ago mobile phone were large, had bad battery capacity and you could only call and send text messages with them. Now mobile phones have good cameras, you can make video calls, send multimedia messages, surf on the internet, listen to music and some phones even support wireless networks and voice over IP. Mobile phones are no longer only for businessmen or wealthy people; in fact in western countries almost anyone can afford a mobile phone. In 2006 the mobile penetration in Norway was 108% according to Norwegian Post and Telecommunication Authority, and the rest of Western Europe is expected to reach the 100% mark during 2007.

With this high availability of communication technology one should think that it would have become easier to fulfil people's communication needs. But in fact research shows that people find it harder to communicate (PitneyBowes 1998). One of the reasons for this is that it is harder for the user to choose the right communication method for the message they want to communicate (Ibid).

As each new communication technology has become available more and more people have become dependent upon them in their workplace or in their everyday life (Ljungberg 1996). It is not the use of these new communication technologies that is the problem, but rather the amount of communication that people are exposed to. The problem becomes clear when more than one communication technology is used at one time. This has become common in most workplaces, where we have a work phone, mobile phone and email available. If you receive an email and your mobile phone rings at the same time while you are in a conversation on your work phone you are exposed to what Ljungberg call “Interaction overload”.

This can often be because the people that want to contact us cannot see us, and therefore they can not know our context, presence or availability. If we had the possibility to share this information with the people that we communicate with it could become easier to choose the right time to communicate and the right communication method.

Earlier there has been no standard for the sharing of presence information, but now the work of the IETF’s Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions (SIMPLE) workgroup have started to show results. The new framework developed by this workgroup is based on the voice over IP protocol SIP, and is designed to be used by any kind of terminal with any kind of network connection. Be that laptop or mobile phone, GSM or wireless network – when the terminal you use supports this framework you will be able to share and receive presence information easily with friends and colleagues.

It is these new developments in technology, with new standards and mobile phones with voice over IP that I want to explore in this thesis. By taking advantage of these new developments in communication will hopefully become easier and less time consuming.

## 1.2 Problem area

This thesis is about the field of human-computer interaction and the mobile users need to communicate his presence, and the technology to support this.

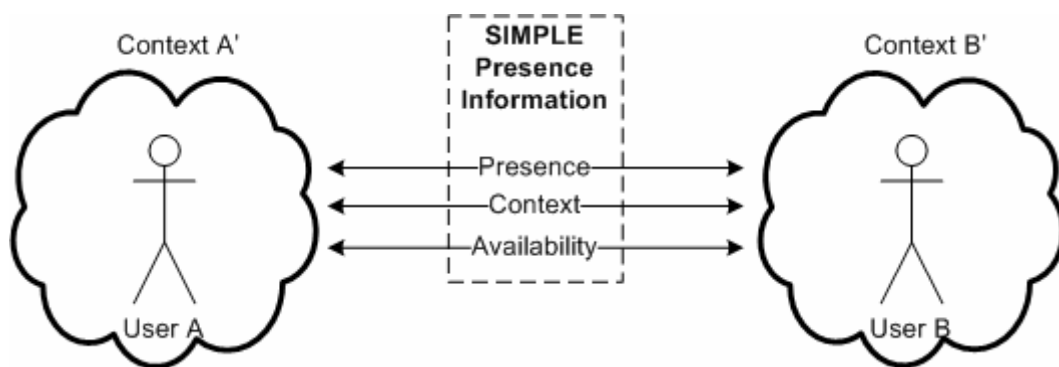


Figure 1 - Problem area

The problem area of this thesis covers the terms communication, mobility and technology. More precisely we can say that the problem area focus on the technology that mediate communication for the mobile user.

## 1.3 Problem definition

The focus of this thesis is to explore how the new SIMPLE framework can be used to provide support for presence information sharing and see how presence information can be used to increase the availability as well as provide new services for voice over IP for mobile users. To narrow down the focus I have defined the following problem definition:

**How can the SIMPLE be used to share presence information among mobile users?**

*- Problem definition*

To further specify the area of research I have defined the following goals and sub-questions:

1. Create and discuss a prototype based on the SIMPLE framework
2. Create and discuss routing alternatives using voice over IP and GSM based on presence information
3. How will SIMPLE be able to help avoid the problems with interaction overload?
4. How will SIMPLE handle privacy issues related to presence information sharing?

By developing a prototype setup using the SIMPLE framework I would like to show how it is possible to deploy the sharing of presence information using an open source standardized framework. And with this technology available I will present how the user will be able to control his communication while still being available. The user will be in control of when he wants to communicate and how the communication will be done. This will also aid the users that want to communicate with this person as they will see the communication methods that the receiver accepts in his current context.

It is crucial to understand when reading this thesis that the notion of *presence information sharing* is understood as described in the SIMPLE framework where presence information covers *context*, *availability* as well as *presence*.

## ***1.4 Overview***

### **Chapter 1: Introduction**

Here I describe the motivation, problem area and problem definition for this thesis.

### **Chapter 2: Methodology**

In this chapter I describe the methodologies used for the research this thesis and explain why they were chosen. The three methodologies used in this thesis are *literature studies, use cases and scenarios* and *prototyping*.

### **Chapter 3: Communication theory**

Communication theory is an established field of study. I will present some well known models for discussing communication, as well as a modified model to better explain mobile communication in relation to semantics and context. I will also present central concepts for presence information sharing such as: *presence, context, awareness* and *interaction overload*.

### **Chapter 4: Communication technology**

Communication technology is a fast developing field. Here I will present the relevant communication technologies for mobile users. I will also explain the basics of the protocols for voice over IP and SIMPLE.

### **Chapter 5: SIMPLE prototype**

In this chapter I first present four use cases and scenarios for using mobile phone with both voice over IP and GSM supported by presence information. I describe the architecture for the prototype as well as the server and client that were chosen for the prototype setup. The configuration of the SIMPLE framework on the server is explained briefly and I present the test results from the prototype with presence information sharing.

**Chapter 6: Discussion**

The findings from my studies of both the SIMPLE framework and the prototyping will be discussed in respect to the presented theoretical framework and the goals and questions lined out in the problem definition.

**Chapter 7: Conclusion**

Here I present the conclusions to be drawn from the research done in this thesis. I also present two fields for future research on this subject.



## 2 Methodology

I have chosen to use Galliers (Galliers 1992) approach and method thus using the distinction between approach and method. This is where approach is the more generic concept which describes “a way of going about one’s research”. The approach allows for the employment of different methods and techniques. In keeping with this distinction I have used one approach and three different methods. In my approach I have chosen to use the prototype. My methods are *literature studies*, *use case and scenarios* and *prototyping*. I will discuss as to why they were chosen and also what their characteristically innate limitations are.

The work with this thesis has spanned over a period of twelve months. In the beginning the work had a high grade of uncertainty. The problem area and definition was unclear. To resolve this I used literature studies to help narrow down the problem area and find a fitting problem definition.

But still there has been some uncertainty. Since the technology I am studying in this thesis is still in development and not in use in any commercial software there have been many challenges. By selecting good methodologies for studying a new technology I have managed to get a good understanding of the basics of this technology and also been able to explore how this can be implemented technically.

### 2.1 *Literature studies*

The first stage of my preparations for this thesis was literature studies. These studies were essential to get an understanding of the technology that I was exploring, as it was so new. Through reading of Request for Comments (RFC) documents, and academic papers on the topic I have been able to get a basic understanding on how this new technology should be implemented.

I have further read about different communication theories, to better understand what impact the use of this new communication technology will have on the way we communicate with each other.

Common for all the literature that I have read for this thesis, is that as a general rule of thumb I have only read material that has been submitted and published in academic forums as well as documents published by the standardizing organizations on the internet such as the Internet Engineering Task Force (IETF).

## ***2.2 Use case driven development and scenarios***

*"Use cases, stated simply, allow description of sequences of events that, taken together, lead to a system doing something useful" (Bittner and Spence 2002)*

A use case defines the interactions between external actors and the system under consideration to accomplish a goal. Use cases treat the system as a black box, and the interactions with the system, including system responses, are perceived as from outside the system. This is a deliberate policy, because it forces the author to focus on what the system must do, not how it is to be done, and avoiding the trap of making assumptions about how this functionality will be accomplished.

Scenarios are a well-known means to "capture valuable information about how users actually go about doing their work". The usage of such scenarios enables us to find characteristic elements of such work and "orients design and analysis toward a broader view (...)" (Carroll 1995). Deriving requirements from scenarios is a task in which scenarios are "used as sources of information about the objects in the domain and how they interact" (Carroll 1995). By carefully examining interactions, artefacts and state transitions, analysts can identify mechanisms and their characteristics that can be regarded as requirements for the support of the processes described in scenarios

Use cases and scenarios are used in this thesis to describe how we by using the new possibilities provided by the SIMPLE framework can use presence information to make the choice of what communication technology to use.

### ***2.3 Prototyping***

Prototypes in the software industry are different from the prototypes made in other industries, such as the car industry. In most other industries the prototype serves as a first sample of a product that is going to be mass produced. However in the software industry reproducing a program is not a technical problem such as in other industries. A software prototype is therefore different from other prototypes because it does not only simulate the finished product, but it actually demonstrates the practical use for it.

The uses of prototypes are much the same as those found other industries. Prototypes are introduced to discover development problems, to serve as a discussion basis between the user and the developers and as a basis to get practical experience through experimentation.

There are three main approaches to prototyping:

- **Throw-away**  
The prototype is built and tested. The design knowledge gained from this exercise is used to build the final product, but the actual prototype is discarded.
- **Incremental**  
The final product is built as separate components, one at the time. There is one overall design for the final system, but it is partitioned into independent and smaller components. The final product is then released as a series of products, each subsequent release including one more component.
- **Evolutionary**  
Here the prototype is not discarded and serves as the basis for the next iteration of design. In this case, the actual system is seen as evolving from a very limited initial version to its final release. Evolutionary prototyping

also fits in well with the modifications which must be made to the system that arise during the operation and maintenance activity in the life cycle.

For this thesis I will be using the throw-away prototyping method. I chose this method since it is not intended to make a fully usable program, but yet explore the different design options and understand the technical implementation.

## 3 Communication theory

In this chapter I describe three communication models which give a good basis for understanding the complex elements which constitute communication. The concept of mobility is also introduced and different types of mobility are presented. I introduce key concepts for presence information sharing such as; *presence*, *context* and *awareness*. Finally I describe the theory behind interaction overflow.

### 3.1 *Background information*

The ability to communicate is one of the most typical characteristics of humans, where natural speech has been the most common form of face to face communication. Methods for communicating over long distances have existed for a long time; the earliest forms of long distance communication were by using drums or smoke signals. These were the first forms of telecommunication, which were defined as communication over distance, and this is one of the focuses of this thesis.

The choice of what telecommunication technology to use can be vital for the success of the communication; this is also one of the focuses of this thesis.

### 3.2 *Communication model*

When analyzing communication it can be useful to use a model or a framework as a basis. A model attempts to capture the essence of a real life phenomenon and present it in a more simple way. By being simple it becomes manageable and helps us grasp what is really going on. The model should also provide us with the framework for discussing and understanding this phenomenon. A model represents some ideas, articulated by a theory or hypothesis (Hermansen 1965).

The quest for a communication model is not new. The Greek philosopher Aristotle did in his work *Rhetoric* present a communication model containing three factors:

- The speaker
- The speech
- The audience

Which also can be written: “*Who says what to whom?*”

At the time this was written the communication methods were limited considering the options we have today. The array of different methods to choose from has grown considerably and so we need a model that reflects this added complexity. Today we have the telephone, e-mail, chat and videoconferencing amongst others. Several communication models have been proposed that take this wide choice of communication methods into account.

A well known communication model that is often referred to is Harold Lasswell’s model from 1948. The model describes each element in a communication process (Lasswell 1948).

“*A way to describe an act of communication is to answer the following question: Who says what through which channel to whom with what effect?*”

Harold D. Lasswell, 1948

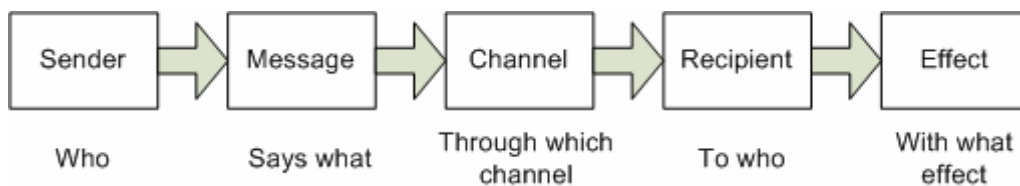


Figure 2 - Lasswell's communication model

Lasswell’s model identifies five important factors: the sender, the message, the channel, the receiver and the effect. Compared to Aristotle’s model this one is more detailed because of the introduction of the channel and the effect. Where as

Aristotle's model was limited to only one communication channel this one includes a component with limitation and possibilities on how the communication is conducted. The effect can be seen as an attempt to put communication into context. As communication often has a goal and purpose that we need a component to measure, say how successful the communication was. This is what the effect component evaluates.

Shannon Weaver's more technical contribution to communication research can be used to look at the problems caused by a physical carrier, whilst also allowing us to look at other factors, introduced after the model had been widely used to describe and analyze communication problems in more general terms. According to Shannon Weaver's model communication can be disassembled, which allows signals to be transmitted and then reassembled at the other end (Shannon and Weaver 1963). This is arguably the most important aspect of his contribution, as the model has been widely used to argue for the encoding and decoding process, which allows for extending noise to be an influencing factor in the communication process as a whole.

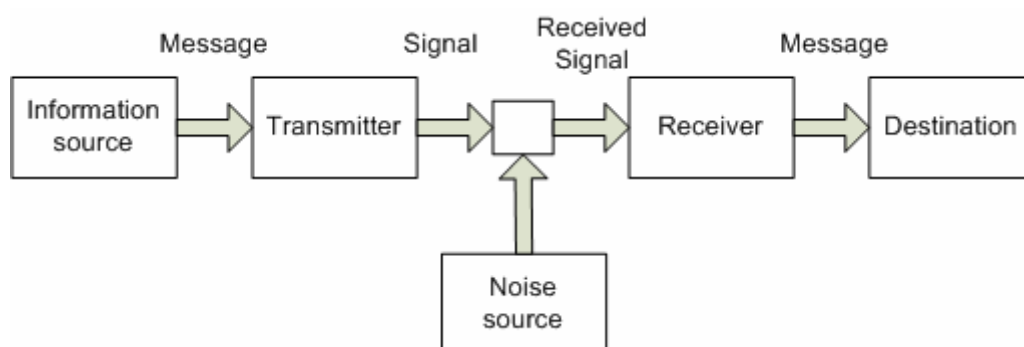


Figure 3 - Shannon-Weavers' communication modell

The additions to the model support and therefore allow us to discuss the crucial question in communication studies:

“To what extent does the message received correspond to the message transmitted”

The Shannon-Weaver model is more often amplified according to the feedback factor, though it is not drawn into the model itself. Feedback allows for us to look at communication as a continuing or ongoing process, not just the pursuit of trying

to get a message to the destination from the source. Founded on the feedback you receive from the recipient you may adapt, adjust or change the communication process in order to increase the chances of the communication being successful.

The Shannon Weaver and the Lasswell model are criticized on the basis of them both being transmission focused models of communication, whilst not addressing how the message is assembled in the beginning. Thus they are transmission models that do not address meaning. In order to address meaning, the models must consider context, as acts of communication may be experienced with a high degree of variation from one context to another.

### Contextual communication model

Olsen's contextual communication model borrows much from the Shannon Weaver model and the Lasswell model but in addition underlines the importance of factors such as semantics and context (Olsen 1999).

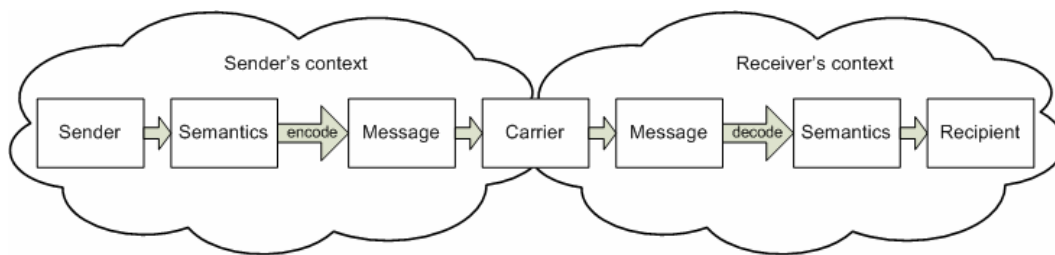


Figure 4 - Contextual communication modell

Olsen's sender in his contextual communication model is the equivalent of the sender in Lasswell's model, as the communication starts with the sender. The sender is a person or persons that have the shared goal of engaging in the act of communication.

The sender has something he/she/they wish to communicate. This something is the meaning. Olsen labels this as semantics. Olsen asserts that the process is critical and that the semantics is transformed into a message that the recipient (should) be able to receive. The encoding process is too responsible for giving the message its formatting. A message can take many forms, say written in English or Norwegian. It may be a recorded voice message in German or a written message using Arial Unicode MS or a handwritten note. Formatting could include images coloured/uncoloured or say drawings in various formats or plain



text. The carrier is the 'what' which will transmit the message from sender to recipient. The Shannon-Weaver model in its original form can be used for discussing this sub-process.

Olsen also states that communication takes place in context. The sender and the recipient both belong to a context. Olsen draws the two contexts as overlapping each other in order to illustrate that they are not identical and we should note that the degree to which they overlap is important in determining if the communication is successful. Olsen also notes that even though it is not explicitly drawn into the model, the feedback and noise are still influential factors.

### ***3.3 Mobility***

Mobility is defined as the ability and willingness to move or change (Dictionary Thesaurus). The type of mobility we look into here is in the technological context.

We can divide mobility into five different categories. They are terminal mobility, terminal portability, personal mobility, application mobility and session mobility. The first three are described in (Audestad 1992), while all except terminal portability are described in (Thanh 1997).

- **Terminal mobility**  
This type of mobility covers the ability of a terminal, while in motion, to access telecommunication services from different locations, and the capability of the network to identify and locate that terminal. A mobile phone is a good example of such terminal.
- **Terminal portability**  
This is the ability to move a terminal from one location to another and still being able to connect to the network through an access point. The access point could be fixed as well as mobile. A laptop is a good example of such terminal.
- **Personal mobility**  
Through unique identification the user can make them self available anywhere on any terminal. The capability of the network to provide services in accord with the user's service profile.

- **Application mobility**  
The ability to transfer a computer process from one machine to another, also while it is processing. Typically used in distributed computer systems.
  
- **Session mobility**  
This type of mobility is defined as an extra service on top of the above mentioned types of mobility, except Terminal portability. With this type of mobility the active session is not interrupted when terminals, persons or application change their location. An example of this is a call transferred from one terminal to another.

In this thesis I will focus on personal mobility, the receiving person is what matters independent of location and terminal.

### ***3.4 Presence***

Presence is defined as the fact or condition of being present, and it is this sense of being present that is used in collaborative communication (IJsselsteijn, Ridder et al. 2000). Lombard and Ditton (Lombard and Ditton 1997) defined presence as the "perceptual illusion of non-mediation", when the individual fails to perceive the medium throughout a technologically mediated experience.

Synonymous with presence are other terminologies like synthetic presence, virtual presence, ego presence and telepresence, which all refer to the same phenomenon of being in a mediated environment (Draper, Kaber et al. 1998). Because of its psychological nature presence is relevant in an attempt to evaluate human experience in a virtual environment (IJsselsteijn, Ridder et al. 2000).

Some researchers have speculated a great deal around the role of presence in the real world. Our perception of presence in the physical world is such a normal phenomenon that we seldom consider it and only really ever feel it. (Huang and Alessi 1999). Damasio (Damasio 1994) use what they call "the sense of being" as an example of a rather neutral innate feeling that comes from an inner bodily physical state rather than an emotional state. These inner physical states however seem to have an effect upon the emotional state.

As a result of this presence in the real world is a “basic state of awareness” (Biocca 1997) rather than a stable factor within our consciousness that isn’t under any continual change (Freeman, Avons et al. 2000). When we process this intermediate stimulus, emotional feelings play a great role.

The term telepresence was coined by Marvin Minsky i 1980 (Minsky 1980) and refers to the phenomenon where a human operator develops a feeling of physical presence in a remote location through interaction with a human interface, for example through the users actions and the following noticeable feedback the user receives via the telecommunication technology.

Early on Johnson and Corliss (Johnsen and Corliss 1971) discussed the importance of designing screens and controls that can “help the operator to communicate his or her presence” in a distributed remote work environment. From the early 1990’s the subjective sensation of presence in distributed and group based work environment has been focused on, in comparison to, various media and the most remarkable of these has possibly been Virtual Environments (VE).

### **3.4.1 Physical and social presence**

In an attempt to unite an assortment of six different conceptualisations of presence found in literature Lombard and Ditton (Lombard and Ditton 1997) defined it as ”perceptual illusion of non-mediation”, By this they mean the degree to which a person fails to address the existence of a medium in the course of a technology based experiment. The conceptualisation Lombard and Ditton identified can be grouped into two categories, physical and social.

The physical category refers to the feeling and also being physically located in the same place while the social category refers to the feeling of being together (and communicating) with someone. We can here ask ourselves to what extent it is relevant to group these to categories under a common definition as many of the communication aspects that are central in social presence are unnecessary to be able to establish a feeling of physical presence. A medium can without doubt offer a high level of physical presence without having the capacity or the possibility to send mutual communication signals at all. Conversation wise, one can feel a form of social presence or nearness to whom we communicate with, by

the use of applications that can offer just a minimum amount of physical representation, which is the case when using for example telephone and discussion boards on the internet. This indicates that there could be a relevant division made between physical and social presence, though they are relative to each other. It is most likely that there would be a certain number of common deciding factors, for example the immediateness of the interaction that is relevant to both the social and the physical presence. As a matter of fact applications like video conferences or shared virtual workplaces are based on being able to provide a mixture of both physical and social presence. This indicates that as technology eventually transports nonverbal shared effects, such as eye contact and body language, that the social presence will therefore rise.

### **3.4.2 Measuring presence**

Research carried out on the topic of presence is still in the early stages. At the present time there does not exist any generally accepted theory on presence. The technological advances have only recently reached a level which is motivating and enabling enough to allow us to undertake a systematic investigation of presence. At this moment in time we are still missing an exceptional paradigm to determine presence. Consequent to this there has been a large number of presence measurement tools suggested. A solid, robust and usable measurement of presence could offer visual-, user-interactive-, and content developers a tool to be able to evaluate a media within a user centric design approach, by making it possible for them to identify and to test these factors that can produce the optimal level of presence for the user. A durable and stable presence indicator would allow for the establishment of equivalent classes for maintenance of presence levels whilst still allowing us to weed out factors that are contradictory (Ellis 1996). In addition to this, a well functioning presence measurement tool would help experts dealing with human factors to take a closer look at the connection between presence and work efficiency and also help us to understand better the general comprehension of experiencing presence in the real world.

### **3.5 Context**

One can break down the complex universe that constitutes context into two categories: One category regarding the user and one category for the physical

surroundings or environment. A context-model by (Schmidt, Beigl et al. 1999) proposes the two categories *Human factors* and *Physical environment*.

### **Human factors**

- User
- Social environment
- Task

### **Physical environment**

- Location
- Infrastructure
- Conditions

The information we have on the user will influence on a basic level how we choose to communicate our message. *Human factors* deal with information we have on the user. This includes their social environment and task. User-information comprises of the users mental capabilities and mental state and also physiological conditions and habits. Looking at information concerning mental capabilities we are given insight into the user's ability to read and comprehend a specific language or to be able to decipher mathematical equations that are articulated/communicated in the communication. Ignoring the above information will inevitably lead to the communication not being able to be understood by the recipient.

Mental state tells us to the extent to which a person is tired, or mentally engaged with other tasks. The semantics found to be important enough to be communicated will be affected by this information. Information as to whether the recipient is tired or asleep will result in the caller not calling the recipient unless it is a necessity to do so.

You can adjust the communication process through having information regarding physiological conditions. They are of importance as they contain information for instants if the recipient is hearing impaired or blind.

Factors such as formal structures and informal structures are part of the social environment. This is the relationship between the sender and the receiver. Formal structures such as rules and procedures may decide how a task should be carried out. Informal structures may override the formal structures. Informal structures

are social factors. An example of this could be knowing that your manager would prefer you send emails to him/her than to call. Knowledge of the social environment is sometimes imperative in understanding the communication process and being able to communicate successfully.

The *task* includes information about what a person aims to communicate. This allows us to be able to decide if the information we wish to communicate is of relevance for the receiver. An example would be to not invite your manager to a round of golf if you know he/she does not enjoy golf.

*Physical environment* includes all information about the physical not just geographical location of the person. The location shown says where the person is. This may be absolute information from a GPS or may be what is called relative locations like “at work” or “at home”. These two locations have different characteristics regarding infrastructure and conditions.

Infrastructure is far from omnipresent for mobile users. *Infrastructure* is of considerable interest to the mobile user when wanting to know something about the infrastructure of the receiving party and would be of value when trying to establish what form of communication to use. Information regarding infrastructure can tell us about access to (or lack there of) electricity, telephone, internet or computer access. Infrastructure gives insight into internet capacity/connection. This can be vital when choosing to send an email either including large attachments or emails excluding large attachments if we have knowledge that the recipient would have to download sent information over a low-bandwidth connection. Access/availability of software too comes under infrastructure. Infrastructure fashions which carriers, encoders and decoders that are available and the quality of service received. An example of this would be if the application Adobe Acrobat is not part of the infrastructure we will not be able to communicate messages containing Acrobat (PDF) attachments.

*Conditions* include forecast information regarding weather such as wind speed and rain mass. Environmental information such as noise from surroundings is also part of conditions. These properties give information about devices that are useful or unsuitable in the current context. An example would be that mobile phones are not suitable in an excessively noisy location..

### 3.5.1 Context sharing

We need a process called context sharing in order for a communication process to be successful. Both the sender and the recipient need to know something about each others context. The applications used for communication today do not fully support context sharing. Most communication applications do not have any information about the recipient's context. There have been suggested a platform for contextual communication that involves a Context Register (Herstad, Thanh et al. 1998) and (Ljungberg 1996). The idea has been put forward that these solutions can be implemented as a client-server solution. The client would reside on the user's terminal and would regularly update the server. When someone wants to communicate with a person the client application would lookup the receiver in the Context Register. This look-up would reveal preferred carrier and also mode and formatting, and in addition to this, reveal information regarding the recipient's capabilities. For this to be successful and by that we mean that people actually would use the application, the information contained in the Context Register would have to be accurate. Manual updates by users have a tendency to be done sporadically. If this were to be the only way of updating the register the information would soon become inaccurate and outdated. Without accurate information people will not use the application. There have been several schemes for automatic updates suggested; one scheme is to use carious sensors in conjunction with an application that automatically updates the Context Register. A sensor could notice that you are in a noisy environment, and the application would then report to the Context Register that you should not receive phone calls. A GPS enabled sensor could report your location, and the application would update the Context Register with the country you are currently in and the time zone to which it belongs.

One concern with this scheme is the issue of privacy. Making information about your context publicly available is not always wanted. One might not always be interested in your friends, family and colleagues always knowing where you are for instance. There will have to be some options to choose what information you would like to share, and also different settings for different type of contacts. One might be willing to share an almost exact location with your friends and family, while your co-workers only would see which country you are in.

With the technology that is being explored in this thesis, the SIMPLE framework, many of the ideas from the Context Register and its suggested schemes is being

made openly available. SIMPLE is highly configurable and also takes privacy in to account.

An example of a communication-service that includes a context register is Skype. This is an internet service that provides voice over internet communication, as well as instant messaging.

The application provides a contact directory where you can add your contact in a contact list. It also includes some support for context sharing. The user may select from a menu what kind of availability to share with its contacts. As you can see by the illustration bellow the list is by no means comprehensive as to describing your context, but the application also provides an option to enter your current location and time zone. All these updates are manual, but it also offers the possibility to automatically set your availability to “Away” if you leave the computer unattended for a certain amount of time.



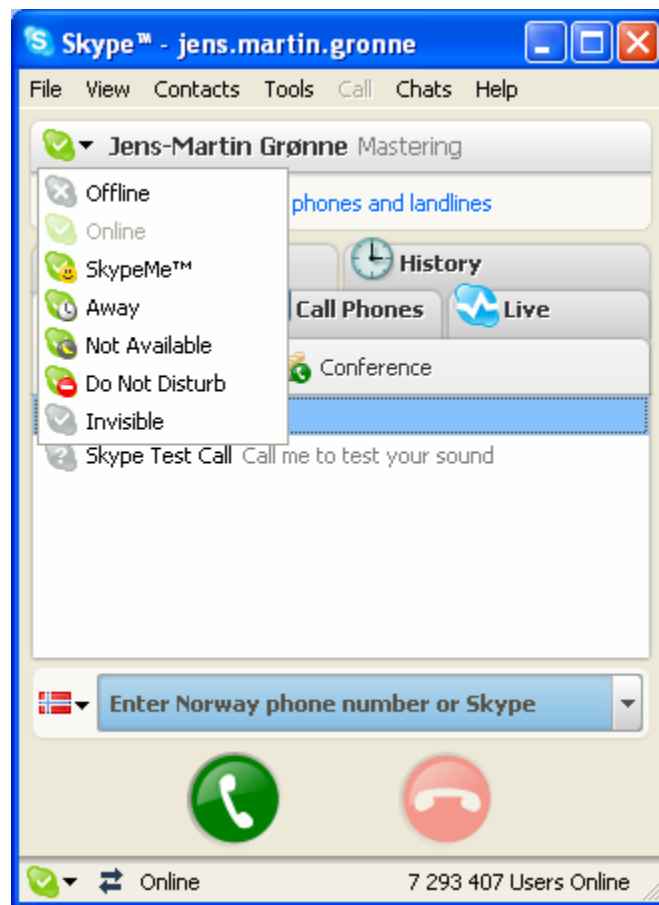


Figure 5 - Skype - a context-aware application

### 3.6 Awareness

A good and often used definition of awareness is:

*"... Usually the term awareness refers to the awareness of the presence of other people"*

It is generally accepted that the communication between people is greatly reliant upon the context in which it takes place.

Participants in a face-to-face conversation adjust their behaviour according to who is present, the other participant(s), the social setting and the degree of intrusiveness, only to mention a few variables. As a consequence of the contexts important role in natural communication there has been carried out a great deal of work looking into trying to understand and add context to group-technological communication. Most of this work has taken place in the field of computer-supported collaboration systems (Erickson, Smith et al. 1999), (Gutwin and Greenberg 1996), (Karsenty 1997), and in later times around chat and instant messenger systems.

There has been a general focus on gathering information about people's status before the communication finds place. This information has included to what degree the person wishes to be contacted, if they are in front of their computer, busy or in a meeting, or if the person is in another country even. In the same way that social and physical intimations can make easier the negotiations that start a face to face conversation, (Clark 1996) this information may possibly predict the likelihood of technology based communication being accepted and in what shape or form it will take in the future. Communication of this type of information is referred to as Personal Presence or Awareness.

In contrast to research undertaken in personal presence and awareness information by PC orientated researchers a telephone has little of this kind of information. The busy signal for instance, indicates an extreme form of unavailability and it is common for another call to be placed before the participants can start to 'negotiate' again, as the person who places the call does not have sufficient information about the person they wish to contact. This can cause regular interruptions where the person is called at times or at places where the person has already left (if you look at landlines). The receiver of the call who wishes not to be interrupted can in these cases make him or herself unavailable by turning off their telephone, but they then at the same time risk missing an important call.

Designers and developers of PC based Personal Presence – Awareness have had to relate and adapt themselves in the face of great design challenges. The first being informativeness vs. privacy which can lead to problems when trying to give information on a persons status well enough to give relevant information to others, but at the same time try to avoid conflicts of interest with regards to the persons wishes for privacy.

An example would be video. When video is used as an informative and is an entirely automatic tool used to report personal presence it is normal for the users and other state-run anti-monitoring authorities to report worries and concerns around the right for privacy. As a result of this many have reduced the detail level of personal presence information to reach a higher level of privacy. This is often done by distorting video or sound signals and by reporting presence in a more symbolic way by use of avatars.

The other obstacle is overhead vs. control. Here people maintain their own presence information. Here there is a great need for constant manual updates according to a continually changing presence throughout the day. Grudin (Grudin 1994) considers overhead to be one of the main flaws in most collaborative systems. In an attempt to eliminate the overhead there have been launched several techniques that could offer automatic tracking of status. Examples of these kinds of techniques are video (Bly, Harrison et al. 1993) body movement sensors (Greenberg 1996) and active signs (Hopper, Harter et al. 1993). The problem with these types of informers is that they are registered automatically thus giving the user less control over it. There can be times where the user wishes to hide his or her presence from others and in this way organisations that work to ensure that information about us, that is registered, is not used against us. This is where they come up against areas of great conflicts of interest. With such systems you cannot avoid being contacted based upon presence (An example of this would be not attending a family gathering due to sickness, and then going shopping). We would most likely see these problems occurring if we were to offer personal presence technology for telephone, though it is not given that the same design solutions used for PC based communication would be used for telephone conversations.

### ***3.7 Interaction overload***

Interaction overload is a term used for describing problems concerning collaboration between people. Further distinction can be made by dividing it into two categories; Communication Overflow (CO) and Communication Deficiency (CD) (Ljungberg and Sørensen 1998). Communication that is not wanted in the current context is considered communication overflow and communication that is not done with the right communication method is considered a communication deficiency. These terms will be explained further below.

### 3.7.1 Communication Overflow (CO)

More and more technologies for communication, collaboration and coordination have in recent years become available. These technologies are part of the term Computer Supported Cooperative Work (Grudin 1994).

When technologies like email and mobile phones first have found their way into our everyday life we often become dependent upon them. At first it will not seem like a problem, but as the amount of communication increases the communication becomes unwanted as it takes up too much time and attention (Ljungberg 1996).

Based on communication models Ljungberg has developed a theoretical understanding of the cause of interaction overflow. In summary we can say that interaction overflow occurs when either the sender or the message is unwanted in the current context, or even independent of the context (Ibid).

Based on this understanding Ljungberg has found that communication overflow is caused by one or more of the following scenarios:

1. Context in relation to sender. When communication from a certain person is unwanted in the current context.
2. Context in relation to the message. When the message is unwanted in the current context.
3. Sender independent of context. When we do not want to communicate with a certain person independent of the context.
4. Message independent of context. When we do not want the message independent of the context.

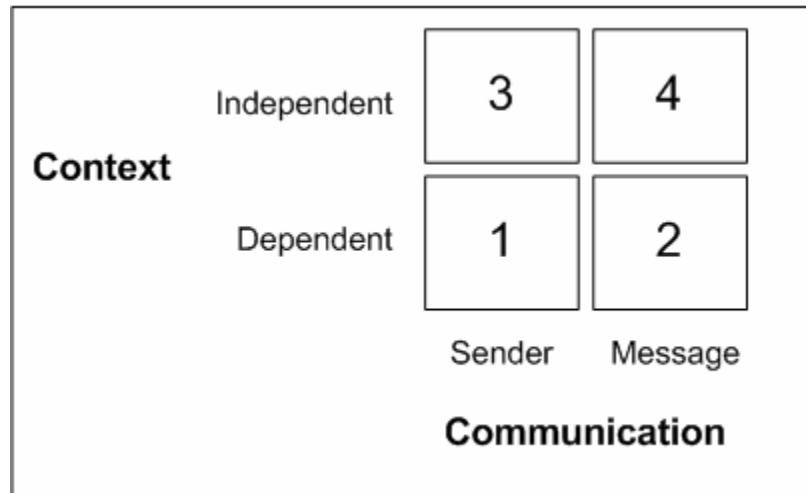


Figure 6 - Communication overflow

### Regulation mechanisms

Regulation mechanisms support the management of communication overflow by distinguishing desired communication from undesired. There are two distinct types of regulation mechanisms: *filtering mechanisms* which support specification of filters that automatically match and direct incoming communication, and which are an integrated feature in some communication applications, and *acknowledging mechanisms* providing information about communication before the user is subjected to it.

### Filtering mechanisms

With filtering mechanisms all incoming communication is filtered before it is presented to the receiver. The filtration is done automatically by predefined rules and without intervention from the receiver (Ljungberg 1996).

A good example for such a filter is virus and spam filters that reside on most mail servers today. The filter is setup with a set of rules that checks for harmful material and unwanted spam, based on the rules the server takes action when it finds email matching the rules criteria.

The major disadvantage of filtering mechanisms is that they do not provide awareness about the communication which has been filtered out.

### **Acknowledging mechanisms**

With acknowledging mechanisms the user has to actively accept the incoming communication. By evaluating the sender the user can choose whether or not to accept the communication (Ibid).

As an example, all internal telephone calls at a company will display the caller's local extension on the telephone of the receiver of the call. As people learn who has which number, this display provides information that could be used to evaluate the desirability of communication before being subjected to it.

The major disadvantage of acknowledging mechanisms is that people must evaluate all communication, and either accept or reject the communication

Accordingly the major disadvantage of filtering mechanisms is the major advantage of acknowledging mechanisms, and vice versa. Therefore, the combination of these two types of regulation mechanisms is potentially a very powerful means for managing undesired communication (Ljungberg and Sørensen 1998).

### **3.7.2 Communication Deficiency (CD)**

Communication deficiency characterizes situations where people are subjected to communication which they are interested in, but where the communication method is undesired (Ljungberg and Sørensen 1998).

To better explain communication deficiency (Ljungberg and Sørensen 1998) has introduced some terms to better explain the phenomenon.

#### **Obtrusive - unobtrusive - communication**

Obtrusive communication is communication that requires the users to react. This type of communication requires the users' attention immediately.

Unobtrusive communication is communication that does not require the users' attention or demands a reaction immediately.

### **Ephemeral – persistent communication**

Ephemeral communication only exists at the time it is conducted and leaves no trace to show that the communication ever has taken place.

Persistent communication is communication that leaves a persistent trace, such as a letter or a note.

A telephone call would be obtrusive since it would require both attention and reaction from the user. At the same time it would be ephemeral because it does not leave a persistent trace with the content of the communication. If the call got transferred to a voice mail the communication would change to be unobtrusive and persistent.

In the figure below we can see that a message delivered discretely that leaves a trace, such as a letter received from the mailman would be considered a persistent and unobtrusive way of communication. An email that makes a sound when it is received and also requires the user to reply urgently would be considered persistent and obtrusive.

When several people are gathered and someone thinks out loud, hums or makes other kind of noises it is considered ephemeral and unobtrusive, since it does not require anyone to listen.

<p><b>Obtrusive</b> <i>Impose obligations to notice and react</i></p> <p><b>Unobtrusive</b> <i>Inconspicuous interaction with no direct obligations to notice or react</i></p>	shouting in a meeting or on the telephone	electronic mail with request to urgently reply
	humming, gazing, thinking aloud	discretely leaving trace, such as PostIt note
	<p><b>Ephemeral</b> <i>Only exist in the flux of unfolding activities and leaves no trace</i></p>	<p><b>Persistent</b> <i>Leaves external trace</i></p>

Figure 7 - Communication deficiency (Ljungberg and Sørensen 1998)

### Choose the right communication method

How we should communicate depends on both the context and the message. If an issue is urgent we normally use the phone, and when an issue is not so urgent we send an email. But time is not always of the essence, for instance if we needed some information from a report, it would be much more useful to get the report on an email rather than having someone read it on the phone.

The choice between ephemeral and persistent communication is often decided by the situation, but by combining the two is often very effective. For instance after a telephone meeting, an email could be sent out that summarizes the discussion in the meeting. This way we use the advantages of both communication methods.

### Handling communication deficiency

People often show signs of whether they like to communicate or not. For instance when closing the door to their office. If they leave the door open they show that they are available for communication, but if the door is closed it can mean that the person is busy and does not want to be interrupted, and it might be better to send an email which is not obtrusive (Schmidt, Beigl et al. 1999).



How the signs are interpreted depends on the culture in the communication environment or if a preference has been made clear by a person that uses these signs.



## 4 Communication technology

### *4.1 Mobile phones*

The mobile phone is

#### **4.1.1 History**

In 1969 the Nordic telecom companies decided to try to establish a public mobile telephone network. This initiative led to the NMT-450 (Nordic Mobile Telephone) standard and was launched for public use in Norway, Sweden, Denmark and Finland in 1981/82. The mobile system was an instant success, and the development of the new NMT-900 standard started in 1983. In 1986 NMT-900 was launched as a complementary system to the already operating NMT-450 system (Grimstveit and Myhre 1995).

The same year as NMT-450 was launched in Norway (1982) the work for a European standard for mobile telecommunications was started. This work resulted in the GSM system. GSM was taken into use in Norway in 1993 (Ibid).

The third generation mobile phone system, UMTS was taken into use in December 2004, but it has not gained the popularity of the two earlier generations of mobile phone systems.

### 4.1.2 Technology

#### Network

The networks that connect mobile phones have had tremendous technological developments the last few years. Today there are three different technologies that can be used for communication with mobile phones in Norway; GSM, UMTS and wireless network (WLAN).

Global System for Mobile (GSM: originally from Groupe Spécial Mobile) communications is the most widely used digital mobile phone system and the de facto wireless telephone standard in Europe. It was originally defined as a pan-European open standard for a digital cellular telephone network to support voice, data, text messaging and cross-border roaming. The GSM network has the best coverage of all the mobile networks in Norway at the moment.

Universal Mobile Telecommunications System (UMTS) also known as 3G, is a high speed mobile communication system. It brings new services to mobile phone such as video calls because of the higher data transfer rate compared to GSM. When it was originally launched it supported transfer rates up to 384 Kbit/s, but with the new development of the UMTS standard called High-Speed Downlink Packet Access (HSDPA) one can achieve a theoretical maximum of 14 Mbit/s in downlink data transfer rate. The UMTS coverage is limited to the areas of the country with highest population density.

Wireless networks also called wireless LAN (WLAN) can also be used for calling from mobile phones. Wireless networks based on the IEEE 802.11 standard can now provide transfer rates from 11 Mbit/s up to 200 Mbit/s, therefore it can be used for voice over IP calls. The coverage of wireless networks is often provided by the users themselves, in their own home or at their workplace. But wireless internet service providers (WISPs) are also providing so called wireless zones in many public places.

#### Terminals

Mobile phones (the terminals themselves) have undergone an even more rapid development than the network its self. The first mobile phones that were available with the NMT network could weigh more than ten kilos and had only one function; phone calls. The phones rapidly became smaller and handier, and with

the introduction of the GSM network more services were added to the phones. They were now able to send text messages and also provide data transfer support. The support for data transfer and internet surfing from the phones became even better when the General Packet Radio Service was introduced to the market.

From the late 1990's and onwards the mobile phone became more of a multimedia device. Phones now have cameras built in, music players, FM radios and the ability to send multimedia messages.

The latest addition to mobile phone is the support of wireless networks and GPS location services. With the support of wireless networks mobile phones are now able to call using voice over IP, which can give the users better sound quality and better prices. The GPS location service can also be used together with the SIMPLE framework to provide an accurate location for the presence information shared with other users.

## ***4.2 Voice over IP***

Voice over IP has become a legitimate communication technology for the public during the last few years and as many as 360 000 Norwegians now use voice over IP instead of PSTN or ISDN (Norwegian Post and Telecommunications Authority 2007).

### **4.2.1 History**

The concept of VoIP (Voice over Internet Protocol) originated in about 1995, when hobbyists began to recognize the potential of sending voice data packets over the internet rather than communicating through standard telephone service. This concept allowed PC users to avoid long distance charges, and it was in 1995 that the first Internet Phone Software appeared. While contemporary VoIP uses a standard telephone hooked up to an Internet connection, early efforts in the history of VoIP required both callers to have a computer equipped with the same software, as well as a sound card and microphone. These early applications of VoIP were marked by poor sound quality and connectivity, but it was a sign that VoIP technology was useful and promising.

VoIP evolved gradually over the next few years, gradually reaching the point where some small companies were able to offer PC to phone service in about 1998. Phone to phone service soon followed, although it was often necessary to use a computer to establish the connection. Like many Internet applications in the late 1990's, early VoIP service relied on advertising sponsorship to subsidize costs, rather than by charging customers for calls. The gradual introduction of broadband Ethernet service allowed for greater call clarity and reduced latency, although calls were still often marred by static or difficulty making connections between the Internet and PSTN (public telephone networks). However, start-up VoIP companies were able to offer free calling service to customers from special locations.

The breakthrough in VoIP history came when hardware manufacturers such as Cisco Systems and Nortel started producing VoIP equipment that was capable of switching. What that meant was that functions that previously had to be handled by a computer's CPU, such as "switching" a voice data packet into something that could be read by the PSTN (and vice versa) could now be done by another device, thus making VoIP hardware less computer dependent. Once hardware started becoming more affordable, larger companies were able to implement VoIP on their internal IP networks, and long distance providers even began routing some of the calls on their networks over the Internet.

Since 2000, VoIP usage has expanded dramatically. While companies often switch to VoIP to save on both long distance and infrastructure costs, VoIP service has also been extended to residential users. In just a few short years, VoIP has gone from being a fringe development to a mainstream alternative to standard telephone service.

#### **4.2.2 Technology**

For call setup the two most common protocols are SIP and H.323. SIP is the Internet Engineering Task Force (IETF) protocol for initiating a two-way communication session, specified in RFC 3261. It was initially designed to be less complex than H.323, but as the standard has evolved it has become increasingly complex. SIP is a text-based protocol, in contrast to H.323 which is encoded. A SIP network consists of several endpoints, SIP Registrar Servers, SIP Proxy and/or Redirect Servers. When a user connects to the VoIP network, the user reports his or her location to the SIP Registrar, which is then able to map SIP addresses to IP addresses when calls are made. An overview of a SIP network is shown in the figure below:

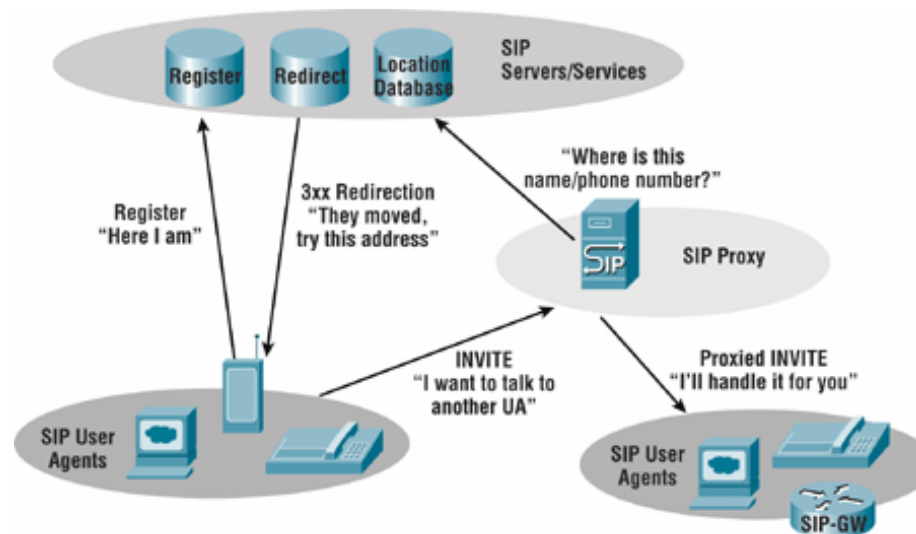


Figure 8 - Overview of SIP network (Cisco 2005)

Call control protocols are used for call setup and termination, and for in-call services such as “call hold” and conference calls. SIP is currently the leading signalling protocol for VoIP.

The figure below shows a basic example of the messages sent back and forth with a SIP initiated call. Here the user “Ann” tries to call the user “Bob” through the SIP Server.

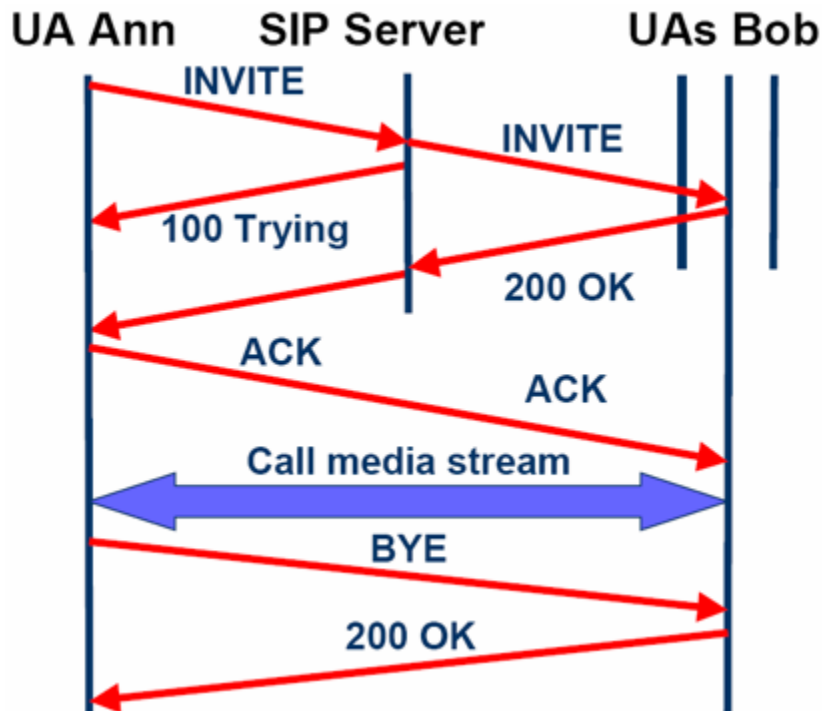


Figure 9 - Basic SIP call

SIP is the foundation that the SIMPLE framework has been built upon, so a basic understanding of this protocol is needed to understand how the SIMPLE framework interacts with the existing SIP platforms.

### 4.3 Fixed to mobile convergence / IMS

Fixed mobile convergence (FMC) is a generic term that embraces terminal device, service and network convergence. That is merging wire-line and wireless networks, service and terminals. With the convergence between the mobile and fixed networks, telecommunications operators can provide services to users independent of their location, access technology, and terminal. The concept of convergence emerges from telecom service providers need to find new revenue stream, reduce their operating expenses and simultaneously invest in future-proof network architectures and technologies. Some service providers are looking for a multitude of new services including mobile and fixed access. The primary goal is concurrent delivery of all media type (Voice, data and video) to an easy to use graphical user interface, independent of access method, terminal and location. The



goal of network convergence is to make all service profitable and enable multiple business models. These goals are related because services that are easy to use become popular and increase revenue. The convergence can then be seen in three aspects or levels, the core network, terminals and services.

#### **4.3.1 Network convergence**

Network convergence means that the same network will be used for both fixed and mobile service and by both operators. This part can be further divided into core network and access network. The goal for the core network is to migrate from separate circuit and packet switched networks to a single unified network that supports the existing mobile and fixed access technology.

#### **4.3.2 Terminal convergence**

Terminals convergence means that terminals should be interoperable across multiple access technologies and vendor networks seamlessly.

#### **4.3.3 Service convergence**

Service convergence is to be able to provide/access new or existing service in both fixed and mobile network independent of your location. This can be composed of one or of combined service, such as videophone. An important future of this is that users can access a consistent set of services from any fixed or mobile terminal via any compatible access point, independent of the access network it is attached to.

#### **4.3.4 IMS**

The fully fixed mobile converged service and network are some years away, but there have been some attempts made. The IP Multimedia Subsystem (IMS) is a standardized Next Generation Networking (NGN) architecture for telecom operators that want to provide mobile and fixed multimedia services. It uses a Voiceover-IP (VoIP) implementation based on a 3GPP standardized implementation of SIP, and runs over the standard Internet Protocol (IP). Existing phone systems (both packet-switched and circuit-switched) are supported.

Part of the IP multimedia subsystem is also SIMPLE. SIMPLE role in this architecture is to provide instant messaging and presence services.

## **4.4 SIMPLE**

In 2001, the SIMPLE working group was formed within IETF (Internet Engineering Task Force) to develop a suite of CPP-compliant (Common Profile for Presence, RFC3859) standards for presence and instant messaging applications over the Session Initiation Protocol (SIP). The SIMPLE activity specifies extensions to the SIP protocol which deal with a publish and subscribe mechanism for presence information and sending instant messages. These extensions include rich presence document formats, privacy control, partial publications and notifications, past and future presence, watcher information and more. Interestingly enough - despite its name SIMPLE is far from simple. It is described in about 30 documents; many of them are still drafts, on more than 1,000 pages. This is in addition to the complexity of the SIP protocol stack on which SIMPLE is based.

To support the SIMPLE protocol are several XML-based data formats that are used for different types of data transfer between SIMPLE clients. I will only present the two most important data formats; PIDF (RPID) and XCAP.

### **4.4.1 Protocol**

A central issue in the SIMPLE framework is how to support presence services. The idea is to provide presence information only to the entities that have explicitly requested it; putting it in SIMPLE words, entities willing to receive presence information of a given entity subscribe to the “presence service” of that entity so that they can be notified when a presence event related to that entity, e.g. coming on-line, occurs.

When a SIP entity (subscriber or watcher) wants to subscribe to the presence service of a remote SIP entity (presentity) it creates a SUBSCRIBE request, carrying the URI of the desired entity. The request traverses normally the SIP network (it passes through chain of proxies as the other requests) until it reaches a SIP presence server, which will generate a response for the SUBSCRIBE request. The presence server, which generates the response is not necessarily the first presence server handling the SUBSCRIBE request; it is also possible that a presence server proxies the request to another presence server, based on local policies decision.

The presence agent (PA) is the logical entity in charge of managing the presence information of a presentity, processing SUBSCRIBE requests, consequently notifying to the subscriber changes in the presence status of the presentity, with NOTIFY requests.

Upon authentication and authorization of the subscription, a PA sends a NOTIFY message to the subscriber including the presence information and whether the request was authorized. Note that it is possible for the PA to send a “faked” NOTIFY message, indicating for example that the presentity is off-line when instead the opposite true. This is useful, since the protocol dictates it to always answer to a SUBSCRIBE with a NOTIFY, even if the request was not authorized. Further NOTIFY messages are sent by the PA to all the authorized subscribers when there is a modification of the presence state.

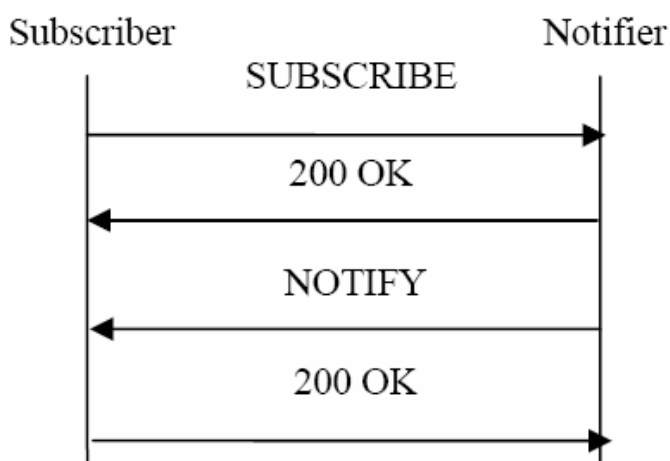


Figure 10 - SIMPLE subscription

Duration of subscription is not permanent, but must be refreshed by the subscriber with a SUBSCRIBE request. This is (among other reasons) to avoid to clog the network with useless NOTIFY messages to which the subscriber is no longer interested. The refresh SUBSCRIBE contains a parameter indicating the duration of the new subscription. In order to close a subscription, it is not necessary to wait its expiration, but either parts can send a request (SUBSCRIBE or NOTIFY) with a duration parameter value set to zero, which causes the receiving entity to immediate close the subscription.

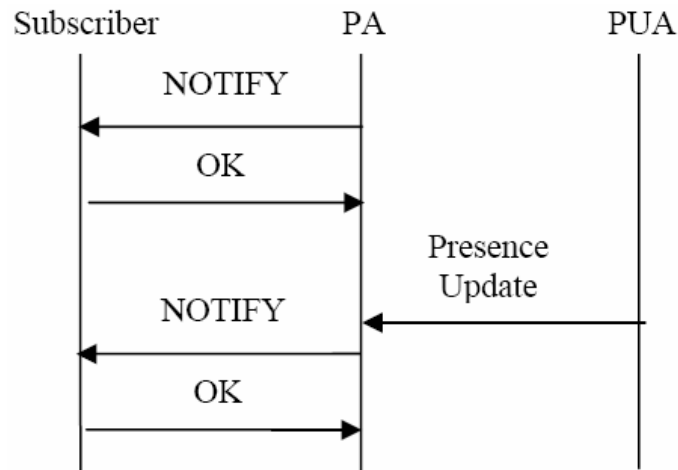


Figure 11 - SIMPLE presence update

The figure above shows how the presence user agent (PUA) on one phone updates his presence status with the presence agent (PA) and how the subscriber to this presentity gets an update from the presence agent. When an update occurs on the terminal the presence user agent on that terminal sends a PUBLISH message to the presence agent containing information about the user of the terminal as well as a PIDF document containing the presence information. This PIDF is stored on the presence agent and is then submitted to all subscribers of this presentity.

#### 4.4.2 PIDF

Presence Information Data Format (PIDF) was created as a common presence data format for CPP-compliant presence protocols, allowing presence information to be transferred across CPP-compliant protocol boundaries without modification, with attendant benefits for security and performance (IETF RFC3863 2004).

The Presence Information Data Format encodes presence information in XML. Using this format the presence information has a hierarchical structure and is fully extensible. Bellow is an example of a PIDF instance document that I will use to describe some of the basic features:

```

<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
  xmlns:im="urn:ietf:params:xml:ns:pidf:im"
  xmlns:myex="http://id.example.com/presence/"
  entity="pres:someone@example.com">
  <tuple id="bs35r9">
    <status>
      <basic>open</basic>
      <im:im>busy</im:im>
      <myex:location>home</myex:location>
    </status>
    <contact priority="0.8">im:someone@mobilecarrier.net</contact>
    <note xml:lang="en">Don't Disturb Please!</note>
    <note xml:lang="fr">Ne derangez pas, s'il vous plait</note>
    <timestamp>2001-10-27T16:49:29Z</timestamp>
  </tuple>
  <tuple id="eg92n8">
    <status>
      <basic>open</basic>
    </status>
    <contact priority="1.0">mailto:someone@example.com</contact>
  </tuple>
  <note>I'll be in Tokyo next week</note>
</presence>

```

The most important elements to notice in this PIDF document include the `<presence>` element which contains the namespace definitions used in the document. In this particular example we notice that two extra namespaces have been defined, one for instant messaging and one custom namespace. Another important element is the `<status>` element which contains status information about the current status of the user described in the document. The `<basic>` element that is below the `<status>` element in the hierarchy can only have the value “open” or “closed”, this status indicates whether or not the user is available for calls or instant messages.

### 4.4.3 RPID

The Rich Presence Information Data format (RPID) is an extension that adds optional elements to the Presence Information Data Format (PIDF). These extensions provide additional information about the presentity and its contacts. The information is designed so that much of it can be derived automatically, e.g., from calendar files or user activity (Rosenberg, Schulzrinne et al. 2006).

The Rich Presence Information Data format expands the Presence Information Data Format with new elements that sets a standard for how to share information about context and availability. These are some of the new elements added:

- **Activities element**  
This element describes what the user is currently doing. Activities can often be derived from calendars, and can hold information like: appointment, lunch, travel and meeting.
- **Mood element**  
The mood element contains one or more elements describing the users current mood. Mood elements can have values like: happy, sad, confused and sick.
- **Place-is element**  
This element describes properties of the place the person is currently at. This offers the watcher an indication of what kind of communication is likely to be successful. The place-is element is divided into two sub-elements; audio and video describing the conditions for this type of communication. The audio sub-element can have values like: noisy, ok, quiet. The video sub-element can have values like: toobright, ok and dark.
- **Place-type element**  
The place-type element describes the current the type of place a person is in. This can be used to indicate what kind of communication that is appropriate. The values that this element can have are defined in RFC4589, which contains values such as: public-transport, theatre, restaurant and many more.
- **Sphere element**  
The sphere element describes the current state and role that the person plays. This element can describe if a person is in work mode, at home or participating in other activities. This information can be used to limit certain groups to see presence information about you when you are not in a sphere concerning them.

- **Time-offset element**

The time-offset element describes the number of minutes of offset from UTC at the person's current location. By having this information the watcher can see the local time at the location where the user is.

#### **4.4.4 XCAP**

Extensible Markup Language (XML) Configuration Access Protocol (XCAP) is defined in draft-ietf-simple-xcap-12 (Rosenberg 2006). This protocol was created to manage the per-user data needed to give an application privacy support and it is supported for use with the SIMPLE framework.

The way XCAP works is very similar to what we see in instant messaging application available today. When you add a contact to your contact list the client on your terminal will issue a SUBSCRIBE request to the presence server to tell it that you want to subscribe to this user's presence information. The XCAP server then checks if this user is already in your contact list on the XCAP server, if it is you are already authorized to see presence information about this contact, if not there will be sent a NOTIFY message to the contact that you are trying to add. This NOTIFY message will ask the client to choose whether you should be able to access his presence information or not. If he grants you access to this information this will be written to the XCAP server so that this process will only have to happen once.

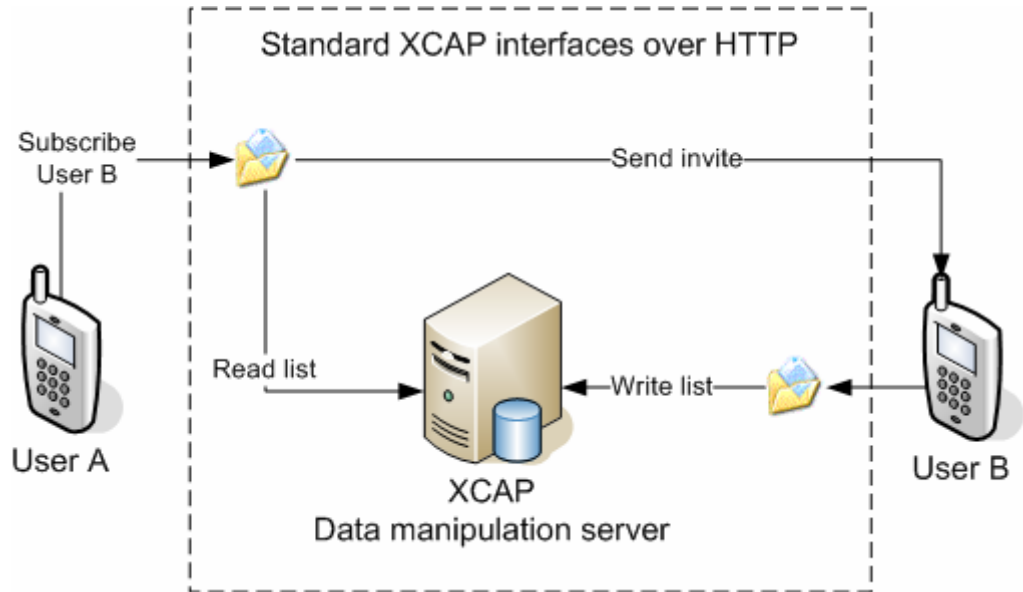


Figure 12 - XCAP authentication

XCAP support standard HTTP for communicating between client and server so no proprietary protocol is needed for XCAP to work. The protocol has also been designed with mobile users in mind. To minimize the use of bandwidth the full resource list is only sent when absolutely necessary. When editing a contact in the resource list only the data needed for that particular contact is transferred to the server. By using a caching system on the client the full list will never have to be transferred when the user wants to look at it, and supporting this caching system is an invalidation function that will make sure that the list is up to date.



## 5 SIMPLE Prototype

In this chapter I will present several different use cases and scenarios that will make the foundation design for my prototype implementation of the SIMPLE framework. I will also present design issues with this type of application for mobile users and present the implementation platform and architecture.

### 5.1 Use cases and scenarios

SIMPLE supports several different communication methods but to narrow down the scope of potential use cases and scenarios to be used as basis for the prototype setup development I have focused only on calls being made from a mobile terminal with support for both VoIP (using WLAN) and GSM / UMTS.

#### 5.1.1 Connectivity based routing

Routing based on which communication technology the user is currently connected with. For VoIP-enabled phone one would have two options either to call over the mobile network (GSM) or to use VoIP.

##### **Scenario**

The following scenario describes how a call could be established between two phones with presence support using the connectivity information in their presence entities.

*Joe is at his office and wants to call his colleague who he knows is on a business trip in China. Because of the high roaming charges that would apply if he called*

*his colleague on the traditional mobile network he wants to call him using VoIP. Looking up his colleague in his contact list he sees that he is actually connected to the VoIP network and he chooses to call him.*

**Use case**

This use case gives a clearer understanding of the steps needed to make a call based on connectivity information.

1. The user opens the phonebook on the phone
2. The phone displays the list of contacts in the phonebook
3. The user navigates to the contact he wants to call
4. The phone displays the selected contacts' status and available communication methods
5. The user selects his preferred communication method and initiate the communication

**5.1.2 Least cost routing**

Routing based on cost. If the user is available on IP telephony the call will be routed on IP to save money. This especially useful when a user is abroad and roaming charges would apply for GSM calls. This case can be quite difficult to implement if the requirement was to actually compare the costs between the price of a GSM call and the price of a VoIP call. If we assume that VoIP is cheaper than GSM if not calling domestic mobile telephone numbers it would be more feasible to implement. The decision of which communication method to use would be taken by the presence user agent (PUA) on the callers terminal, based on defined rules. But the user will also have the possibility to override the method chosen by the PUA.

**Scenario**

*Joe wants to call his friend that lives in Germany but he also wants to save money by calling over VoIP. Therefore he has enabled least-cost-routing on his mobile phone so that the phone will always choose to call with VoIP if it is available. Joe can the simply lookup his friend in his phonebook and press call and the call will be made with the cheapest communication method.*

**Use case**

1. The user opens the phonebook on the phone
2. The phone displays the list of contacts in the phonebook
3. The user navigates to the contact he wants to call
4. The user press the call-button on the phone
5. The phones presence user agent selects the cheapest communication method and initiates the communication.

**5.1.3 Context / availability routing**

Routing based on the context of the receiving party. Context and availability can contain information like location, current activity and current time zone. Based on this information the user will be able to decide whether or not to contact this person at this time. If the user still decides to call despite the fact that the user is not available at the moment, the presence agent will route the call to the receiver's voicemail.

**Scenario**

*Joe wants to call his wife from work but does not know if she is available at the moment. Earlier on the day before leaving home his wife told him that she had a lot of meetings at work that day. To check if she is available Joe picks up his phone and finds his wife entry in the phone book. He sees that she is still in a meeting and the only available communication methods are SMS and to call her voice mail. He decides to send her an SMS and ask her to call him when she got the opportunity.*

**Use case**

1. The user opens the phonebook on the phone
2. The phone displays the list of contacts in the phonebook
3. The user navigates to the contact he wants to call
4. The phone displays the selected users presence information and available communication method.
5. Based on the presence information and available communication methods the user chooses the communication method that is best for what he want to communicate.

### 5.1.4 Security based routing

Routing based on the security of the current connection. When using VoIP enabled phone security becomes more of an issue compared to using the existing GSM network. The GSM network is encrypted and hard to decrypt even for security experts. VoIP enabled phones relies on wireless networks that is also used by laptops, PDA's etc. A wireless network can be open, meaning that anyone can access it without any form of authentication and encryption. In this situation the VoIP call can be monitored by anyone on the same network. Therefore it can be very useful, especially for confidential calls during, to have an indicator for what kind of security that is provided when placing a call between to VoIP enabled phones. This way the user can choose to call with the normal GSM network if the security available for the VoIP call is inadequate.

#### Scenario

*Joe wants to call his boss to give him some numbers that is going to be included in the company's results for the first quarter of this year. Since he knows this information is confidential and must not fall in the hands of people outside the company he wants to make sure that the call is encrypted. When he looks up his boss in his phonebook he sees that a call using VoIP would not be encrypted, so he decides to use the GSM network for this call.*

#### Use case

1. The user opens the phonebook on the phone
2. The phone displays the list of contacts in the phonebook
3. The user navigates to the contact he wants to call
4. The phone displays the list of available communication methods and an indicator next to the VoIP alternative shows the security level of such a call.
5. Based on the information that is going to be communicated the user selects the most suitable communication method.

### 5.1.5 Selecting a use case

For the prototype setup that I want to implement I will only choose one of the use cases. This will simplify the setup, but my choice of use case is also limited by the availability of supporting technology like clients and servers.

The steps involved for the user for making a call in all the presented use cases are very similar. But the presence information that is needed for the different use cases are quite different. The Presence User Agent (PUA) that is the presence agent on the user terminal will have different specification for each of the use cases. The PUA will also need integration with sensors and different subsystems of the phones operation system. This is to get information like location and noise, but also to get system settings like security and call profile (for least cost routing).

Due to the very large programming effort that would have been needed to implement all of these use cases, and the lack of current supporting software for all of the functions I have decided to only implement a setup for the *context and availability* use case.

## 5.2 *Mobile user interface*

When designing user interfaces for mobile terminals there are few issues that need to be taken into consideration in addition to normal user interface guidelines. The first issue is the obvious fact that you are designing an interface for a much smaller screen than in a normal computer environment. The limited screen size limits the amount of information that can be displayed and one must not be tempted to add too much information to any single screen as this will overcrowd the interface and confuse the user (Dix, Finlay et al. 2004).

Another issue that is important to have in mind when designing a mobile user interface, as well as in normal interface design to some extent, is the user's cognitive abilities. By cognitive abilities I mean the user's ability to apply his knowledge when using the interface. For mobile users this means that the interface should meet the users' expectations, the interface should be similar to

the user previous experience with similar systems and services. This way the user would not have to learn to use the new functions added from scratch (Love 2005).

In this prototype we would need to apply these guidelines to be able to design the new phonebook with presence information in such a manor that it will be familiar to the user. Bellow I have included some mock-ups that I have made of such an interface.



Figure 13 - Phonebook with presence

The mock-up above shows the contact list in the phone's phonebook. It is presence based and will indicate communication methods that are available for the selected contact. Communication methods shown are (from right): GSM call, voice over IP call, SMS, e-mail, IM (instant message) and voicemail. On the figure above all communication options are available and therefore coloured, on the figure below only SMS, e-mail and voice mail is coloured and the others are grey and unavailable. This shows the user that the contact is not available for any instant communication methods.

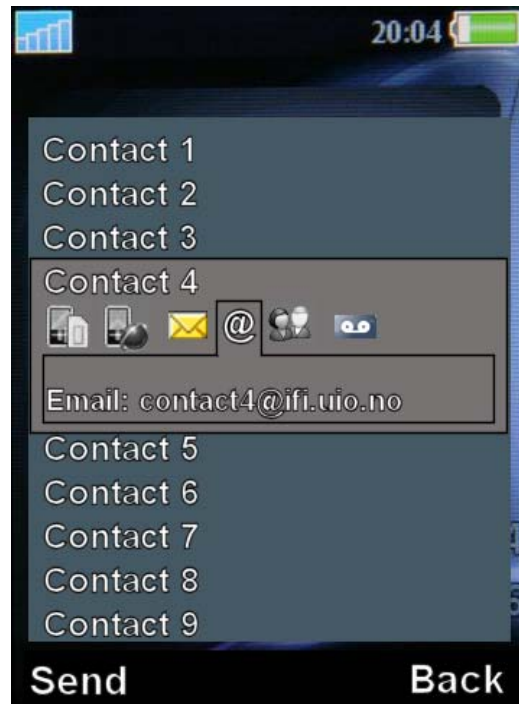


Figure 14 - Phonebook with limited communication methods

Implementing a new mobile interface is a cumbersome task and would take very much time, and since this thesis is about the SIMPLE framework and not about user interface design I find that making such an interface for the prototype is out of the scope for this thesis. In addition to this most of the operation systems that are on mobile phones today is protected by the mobile phone manufacturers and the interface for a phonebook with presence information support would have to be developed by their developers.

### ***5.3 Prototype setup***

The prototype setup in this thesis is a proof-of-concept of the use of the SIMPLE framework for presence information sharing between mobile users. Because of the lack of support of the session initiation protocol (SIP) on currently available mobile phone the prototype setup will only make use of computer based software clients with support for both SIP and SIMPLE.

### 5.3.1 Architecture

The architecture of the prototype setup will consist of two terminals and the “SIMPLE Prototype Server Platform”. The software used to run both the terminal and the server will be explained below. The figure below shows how the information will be shared between the two terminals through the presence server platform.

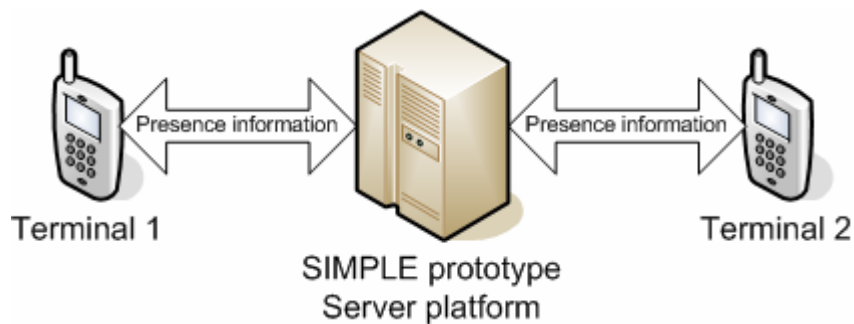


Figure 15 - SIMPLE Architecture

### 5.3.2 Server – Sip Express Router (SER)

As the server in my setup I have decided to use SER. SIP Express Router (SER) is a high-performance, configurable, open source SIP server (iptel.org 2007). It is used by a large number of ITSPs (Internet Telephony Service Providers) throughout the world (Ibid).

SER also includes a presence module that is compliant with the standards defined in the SIMPLE framework. Although not all of the functionality defined in the SIMPLE specification has been implemented it is one of the few SIP servers that actually can be used to test the SIMPLE framework.

All user and presence information in SER is stored in a database backend. This gives the server fast and consistent data access. The database can also be used as



data source for extensions to the presence agent service. With this possibility one can program further logic to the presence agent so that handles communication before it even reaches the user. In my setup I have decided on a MySQL database which is the recommended database backend by the developers of SER.

The figure below shows how SER operates as one server but with two different logical handlers, one for SIP and one for presence. And both use the database to store information.

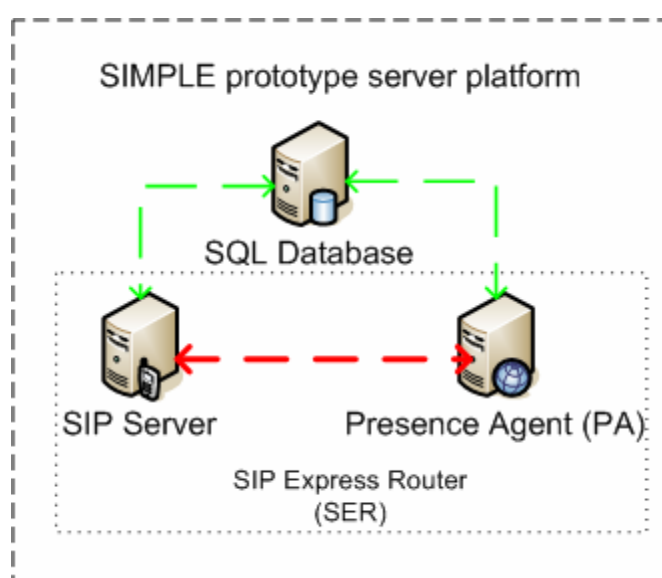


Figure 16 - Prototype server platform

### 5.3.3 Client – X-Lite

As client in the prototype setup I decided to use X-Lite from Counterpath. Counterpath is the developer of one of the most sold soft-phones used for voice over IP. X-Lite is their freeware soft-phone and one of the few fully functional soft-phones that support SIMPLE, although the support is limited.

SIMPLE support in X-Lite is currently at a beta stage and among other things it lacks support for XCAP. Therefore the prototype setup has not been tested with

privacy mechanisms although the privacy mechanisms of SIMPLE have been studied thoroughly.

The screenshot below shows the X-Lite user interface. It is primarily a soft-phone and it has a lot of buttons and functions that you would find on a normal office phone. The contact list that is used for SIMPLE is on the right hand side, contacts are displayed very much in the same way as instant messaging applications such as MSN Messenger. The user can also change his status by selecting on of the predefined options that is show bellow. When this status changes a SIMPLE message will be sent to the presence agent on the server, and the presence agent will in turn notify your contacts of your status change.



Figure 17 - X-Lite Client

### 5.3.4 Configuration

The configuration of SER is contained within one file called “ser.cfg”. The file is divided into four main sections: global parameters, external module loading,

module parameters and routing blocks, which contain request routing logic. The last is by far the largest section.

The insight needed to configure SER is extensive as everything you want it to do must be programmed in the configuration file. It has been difficult to get the presence module (SIMPLE) to work because of the lack of good documentation. But with the help of the developer of the presence module I have managed to create a configuration that does support presence information sharing as well as get the basic routing working. The `ser.cfg` for my prototype is included in Appendix A.

### **5.3.5 Testing**

For testing the prototype setup I have used a packet analyzer to check that the SIMPLE protocol between the clients and the presence agent works as described in the RFC. The only issue I found which departed from the RFC-specifications was how the status of the client was store in the PIDF-document sent from the clients to the presence agent. This has probably been introduced because of the current limitations of the PIDF standard and the lack of support of the extension to the PIDF – RPID on the clients.

Never the less the clients where able to subscribe to each others presence information with the presence agent and were also able to see each others statuses. When the status was updated on one client, it correctly sent out a PUBLISH message to the presence agent containing an update PIDF-document with the new status, and the presence agent correctly sent out a NOTIFY to the subscribing client with the new status. The clients also correctly sent out “keep-alive” subscriptions to the presence agent to notify the server that they were still connected and want updates on this presentity when available.

To test the possibility of routing based on context information I was able to setup a control mechanism on the server that would check the current presence information of the receiving party and route the call based on the status that was found. The control mechanism was setup in such a manor that when a call was placed to a client with the status “Busy” in their presence information, the call would be transferred to a fictive voice mail route.

This routing worked as described above, but it will be much more useful when both the clients and the server accept the new elements defined in the RPID extension to PIDF.

## 6 Discussion

In the introduction of this thesis I presented the problem definition for the work with this thesis. The problem definition was:

*How can the SIMPLE be used to share presence information among mobile users?*

- Problem definition

The problem definition was further specified with four goals and sub-questions. In this chapter I will present and discuss these four subjects.

### 6.1 Prototype

The goal set in the introduction was:

*Create and discuss a prototype based on the SIMPLE framework*

The motivation behind creating a prototype based on the SIMPLE framework was to show how presence mediated communication can be done using an open standard framework.

The prototype became somewhat limited in regards to the initial thought of testing the prototype with mobile phones with support for wireless networks and SIP. Because of the immature nature of SIMPLE, due to the fact that several parts of the standard are still drafts, there are no implementations available for use with mobile phones at this time. However the most important question still remains; is this framework suitable as a standard for presence mediated communication.

### 6.1.1 Results

To look into this I established a test platform based on the open source SIP Express Router (SER), which is one of few SIP gateways with working support for the SIMPLE framework. The configuration of SER was somewhat difficult due to the lack of documentation for the presence module (SIMPLE). But with the help from the developer of the module I managed to get SER running smoothly.

Instead of mobile phones as clients I settled on one of the most popular soft-phone clients on the market; Counterpath X-Lite. This is one of the few clients on the market with support for SIMPLE, although the support is limited. The configuration of X-Lite is easy in its intuitive graphical user interface, and I had no problems connecting it with SER.

With both the clients and server setup I was ready to test how the SIMPLE framework would perform in a basic setup with two clients sharing presence information. For testing the platform I defined the following three questions:

- Is the protocol used by both the server and the clients in accordance with the standards defined in the SIMPLE framework?
- Are the XML documents used to transfer presence data to and from the server in accordance with the standards defined in the SIMPLE framework?
- Is there support for routing based on context / presence information on the presence agent (server)?

The protocols were as far as I was able to tell using a packet analyzer in accordance with the standards defined in the SIMPLE framework, and the communication between the clients and the server worked as intended by the standards defined.

I found one little deviation from the standard in the PIDF-document sent from the clients, but this is likely due to the limitation in the PIDF standard. Neither the clients nor the server supports the new extension RPID yet.

I was able to setup routing based on the presence information stored in the database on the server, although this functionality would be much more consequent if the RPID extension were supported to give standardized information in the presence information data format. But due to the high configurability of SER I was able to get this working.

### **6.1.2 Presence information sharing**

With the successful test of this prototype I feel that I have shown that the SIMPLE framework can be used successfully as a standard for sharing presence information. Several concepts from the communication theory behind the sharing of context are addressed. SIMPLE incorporates many of the ideas of a Context Register as defined by (Herstad et al, 1998) and (Ljungberg 1996). Such as the ideas for the Context Register SIMPLE provides a central server where users can subscribe to each other to receive information about the subscribed contacts current context and communication options.

With the RPID extension to the presence information data format (PIDF) many of the concepts from the theory of context information by (Schmidt, Beigl et al. 1999) are being made available in the SIMPLE framework. RPID adds the support for sharing important context information such as information about the *user* and *social environment*, these are human factors. And it also contains information about the physical environment with *location*, *infrastructure* and *conditions*.

## **6.2 VoIP and GSM routing**

The goal set in the introduction was:

*Create and discuss routing alternatives using voice over IP and GSM based on presence information*

With the use cases in this thesis I have looked into how the new addition of support for wireless network and voice over IP on mobile phones can give new communication options for the users.

Fixed to mobile convergence (FMC) is an important term in the telecommunications industry today. FMC is the way to connect the mobile phone to the fixed line infrastructure. With the convergence between the mobile and fixed line networks, telecommunications operators can provide services to users irrespective of their location, access technology, and terminal. The possibility to use voice over IP on mobile phones provides such convergence. By utilizing the fixed line infrastructure from the central telephone network to the user's home or company (e.g. broadband connection / ADSL) and a wireless network at this endpoint the user will be able to call cheaper and the voice quality will be higher than in the mobile network today. New options on handling the routing of calls can also be given to the user by the deployment of presence information sharing using SIMPLE.

It is these new options that I have looked into and made use cases for. An important aspect for all personal users of telecommunications is the price of the communication. With the help of presence information sharing the presence user agent can determine what communication methods are available and help the user choose the most inexpensive communication method.

For business user there are also other aspects other than the price that can be important to determine before one chooses communication method. The security of the communication is often important when business issues are going to be discussed, and by using the presence information provided by SIMPLE the users will be able to see if the voice over IP communication would be secure before they initiate a call.

With fixed to mobile convergence being deployed rapidly the next few years to come, SIMPLE can help the users choose the right communication method. This is an important issue in both *contextual communication* and with the issue of *interaction overload*.



### **6.3 Interaction overload**

The question raised in the introduction was:

*How will SIMPLE be able to help avoid the problems with interaction overload?*

Interaction overload addresses several issues with communication, issues regarding the context of the receiving party and issues regarding the choice of communication method / technology. The task was to see if some of these issues can be resolved by using the SIMPLE framework for presence information sharing.

#### **6.3.1 Communication overflow**

Communication overflow arises when either the communication from the calling party or the message communicated is unwanted in the current context. Central notions in communication overflow are context, person and message. The following four scenarios were defined for communication overflow:

1. Context in relation to sender. When communication from a certain person is unwanted in the current context.
2. Context in relation to the message. When the message is unwanted in the current context.
3. Sender independent of context. When we do not want to communicate with a certain person independent of the context.
4. Message independent of context. When we do not want the message independent of the context.

From these scenarios we can gather that we need some type of filter, not only based on context, but also based on the person calling as well as the message.

The SIMPLE framework supports this type of filtering. Using XCAP one can define privacy rules based on person or groups. One could for instance have three groups; Family, Friends and Colleagues. Based on these groups one would be able to setup rules for communication methods available to these groups based on context. Let's say you were at the cinema and a colleague tries to call you, based on the privacy group from XCAP the presence agent on either your phone or on the central SIMPLE server would forward his call to your voice mail. But if your daughter called your phone would ring, this based on her membership in the Family group defined on the XCAP server. It is also possible to define rules on a per contact basis, but doing this can often be too cumbersome to administer.

The previous example was only based on the receiving party's presence sharing with the presence agent server and privacy settings provided by XCAP. By sharing presence information with your contacts, so that they can see your current context one could avoid receiving messages that would be unwanted in the current context. This would solely have to be judged by the calling party and might not give consistent results.

One issue with communication overflow that SIMPLE does not handle is the filtering of messages independent of context. There exists no mechanism for filtering the communication based on only the message itself. This filtering mechanism could be wanted to avoid e.g. phone sellers, but would have to be handled by the communication server, for voice over IP this would be the SIP server.

### **6.3.2 Communication deficiency**

Communication deficiency arise when communication is initiated with an unwanted communication method / technology, for instance when someone calls instead of sending an email.

Communication methods have different characteristics that determine whether or not they are obtrusive and if they are ephemeral or persistent. Phone calls are an

example of an obtrusive and ephemeral communication method, while an email normally is unobtrusive and persistent. But how these communication methods are perceived depends on the user's setup.

The presence information sharing provided by SIMPLE is a good solution to the problem with communication deficiency. The receiving party can set its context / location, and based on this the presence user agent (PUA) will set the available communication methods. If you are currently in a meeting the PUA will show your contacts that the only option for contacting you is through email, voice mail and SMS. These options can also be controlled by environmental factors around the receiving party, such as noise level. The sharing of this information would require the new and updated presence information data format – RPID.

The possibility of limiting the communication methods is not always the best solution as this can cause the communication to fail due to the fact that the calling party can not select his preferred communication method. But if one takes full advantage of all the communication methods and technologies support by the SIMPLE framework the probability of such a communication failure is much more unlikely.

## **6.4 Privacy**

The question raised in the introduction was:

*How will SIMPLE handle privacy issues related to presence information sharing?*

From the users' point of view, there is a fundamental trade-off between access to presence data for legitimate uses, and concerns about privacy. Precisely to the extent that people will be able to identify what you are doing, they can communicate with you when the need arises, make their communications more timely and convenient for you. This is the sort of information, however, that users would generally not like to provide to strangers, nor perhaps to managers, or to competitors (Godefroid, Herbsleb et al. 2000).

SIMPLE takes these privacy concerns very seriously and there are several mechanisms to ensure that the information shared only reach the people that the user intends it to. XCAP is an XML-based service that provide authentication for access to a user's presence information. When doing a subscription to a user's presence information mutual accept is required for the authentication to go through with XCAP. This means that the user that is subscribed to will have to accept the request from the subscriber before any information is shared.

XCAP also provides the possibility to group users into different user groups. By separating for example family, friends and colleagues one can setup different rules for what information is shared with the different groups.

There is also the issue of context based privacy (Ljungberg 1996), one might not want to share your presence information with certain groups of people depending on which context you are in. When on holiday and when at home, most people would not like to share their presence information with their colleagues. SIMPLE provides the possibility to set your current "privacy context" using the sphere element of the RPID extension. The information in this element can be used by the presence agent to filter what information different groups defined in XCAP will see.

## 7 Conclusion

Before I conclude I repeat the problem definition of this thesis:

*How can the SIMPLE be used to share presence information among mobile users?*

Much of the work with this thesis has been about exploring how the SIMPLE framework functions in respect to sharing presence information between users, and the theories behind communication and presence information.

Communication does not always come to us in the way we want it to. As a user of a mobile phone our availability makes us easily exposed for interaction overload, overflow and deficiency. Theory suggests that this can be overcome by the sharing of context information by making the users aware of each others context before establishing communication. I have looked into how the SIMPLE framework can be used to mediate the information needed to prevent these types of communication failures. I can say that with the possibilities provided by the SIMPLE framework, users will have a much better basis for selecting the right type of communication method.

An important issue with regards to presence sharing is the issue of the user's privacy. Users will not use presence information sharing if they do not feel they have control over their own privacy. Through my research of the SIMPLE framework I have found that the mechanisms to protect the users privacy extensive. The privacy mechanisms go beyond only securing the users information, and also gives the user the ability to share his presence based on his current context.

Due to the fact that SIMPLE is a new and somewhat immature standard this has been an ambitious issue to explore. I can not say categorically that this is *the* standard for presence information sharing, but I can say that it is a well functioning, complex standard with many features for mediating the sharing of presence information between both mobile and stationary users.

### ***7.1 Future work***

The research I have done on the SIMPLE framework in thesis has been of a technical nature, studying the protocols and the data formats supporting it. But the idea of presence information sharing is highly dependant on the user's interaction with the technology. The study of this interaction could be of high value to see if this framework will be usable in people's everyday life. Interesting issues would be how people use the technology to share their presence, and how they use the presence information they get from other when establishing communication.

Another area that can be further researched is the area of fixed to mobile convergence and how SIMPLE can provide mobile users with presence information to make the best use of the possibilities with these new phones. The study of this area would be dependant upon the introduction of mobile terminals with support for SIMPLE.







## Terminology

### **GSM**

Global System for Mobile (originally Groupe Spécial Mobile) communications, the most widely used digital mobile phone system and the de facto wireless telephone standard in Europe. Originally defined as a pan-European open standard for a digital cellular telephone network to support voice, data, text messaging and cross-border roaming.

### **HTTP**

Hypertext Transfer Protocol (HTTP) is a method used to transfer or convey information on the World Wide Web. Its original purpose was to provide a way to publish and retrieve HTML pages.

### **IETF**

The Internet Engineering Task Force (IETF) develops and promotes Internet standards, cooperating closely with the W3C and ISO/IEC standard bodies; and dealing in particular with standards of the TCP/IP and Internet protocol suite. It is an open, all-volunteer standards organization, with no formal membership or membership requirements.

### **SIMPLE**

Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions is an open standard instant messaging (IM) and presence protocol suite based on Session Initiation Protocol (SIP).

### **SIP**

The Session Initiation Protocol (SIP) is an application-layer control (signalling) protocol for creating, modifying, and terminating sessions with

one or more participants. These sessions include Internet telephone calls, multimedia distribution, and multimedia conferences.

**VoIP**

Voice over Internet Protocol, also called VoIP, IP Telephony, Internet telephony, Broadband telephony, Broadband Phone and Voice over Broadband is the routing of voice conversations over the Internet or through any other IP-based network.

**WLAN**

A wireless LAN or WLAN is a wireless local area network, which is the linking of two or more terminals without using wires.

**XCAP**

The Extensible Markup Language (XML) Configuration Access Protocol (XCAP). XCAP allows a client to read, write and modify application configuration data, stored in XML format on a server. XCAP maps XML document sub-trees and element attributes to HTTP URIs, so that these components can be directly accessed by HTTP.

## Referances

- Audestad, J. A. (1992). "An introduction to Intelligent Networks." Teletronikk **2**: 5-14.
- Biocca, F. (1997). "The cyborgs dilemma: progressive embodiment in virtual environments." Journal of Computer-Mediated Communication **3**(2).
- Bittner, K. and I. Spence (2002). Use Case Modeling, Addison Wesley Professional.
- Bly, S. A., S. R. Harrison, et al. (1993). "Media spaces: bringing people together in a video, audio, and computing environment." Commun. ACM **36**(1): 28-46.
- Carroll, J. M. (1995). Scenario-Based Design - Envisioning Work and Technology in System Development, Wiley.
- Cisco. (2005). "Understanding Packet Voice Protocols." from <http://www.iec.org>.
- Clark, H. H. (1996). Using Language, Cambridge University Press.
- Damasio, A. R. (1994). Descartes' error. Emotion, reason, and the human brain. New York, Avon Books.
- Dix, A., J. Finlay, et al. (2004). Human-Computer Interaction, Prentice Hall.
- Draper, J. V., D. B. Kaber, et al. (1998). "Telepresence." Human Factors **40**(3): 354-375.
- Ellis, S. R. (1996). "Presence of mind: a reaction to Thomas Sheridan's "Further musings on the psychophysics of presence."" Presence (Camb) **5**(2): 247-259.
- An

Erickson, T., D. N. Smith, et al. (1999). Socially translucent systems: Social proxies, persistent conversation, and the design of "babble". Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit. Pittsburg, Pennsylvania, USA, ACM Press.

Freeman, J., S. E. Avons, et al. (2000). "Using behavioural realism to estimate presence: A study of the utility of postural responses to motion stimuli." Presence 9(2): 149-164.

Galliers, R. D. (1992). Information Systems Research: Issues, Methods and Practical Guidelines. Oxford, Blackwell Scientific Publications.

Godefroid, P., J. D. Herbsleb, et al. (2000). Ensuring privacy in presence awareness: an automated verification approach. Proceedings of the 2000 ACM conference on Computer supported cooperative work. Philadelphia, Pennsylvania, United States, ACM Press.

Greenberg, S. (1996). Peepholes: low cost awareness of one's community. Conference companion on Human factors in computing systems: common ground. Vancouver, British Columbia, Canada, ACM Press.

Grimstveit, L. and H. Myhre (1995). "The history of mobile communications in Norway." Teletronikk 4.

Grudin, J. (1994). "Computer-supported cooperative work: History and focus." IEEE Computer Society 5(27): 19-26.

Gutwin, C. and S. Greenberg (1996). Workspace awareness for groupware. Conference companion on Human factors in computing systems: common ground. Vancouver, British Columbia, Canada, ACM Press.

Hermansen, T. (1965). "Formuleringer og bruk av teoretiske modeller."

Herstad, J., D. v. Thanh, et al. (1998). A Component based terminal - the invisible terminal. Ericsson Conference on Usability Engineering, (ECUE 98). Kista, Stockholm.

Hopper, A., A. Harter, et al. (1993). The active badge system. Proceedings of the SIGCHI conference on Human factors in computing systems. Amsterdam, The Netherlands, ACM Press.

Huang, M. P. and N. E. Alessi (1999). "Presence as an Emotional Experience." Medicine Meets Virtual Reality: The Convergence of Physical and Informational Technologies Options for a New Era in Healthcare: 148-153.

IJsselsteijn, W. A., H. d. Ridder, et al. (2000). Presence: Concept, determinants and measurement. IPO, Center for User-System Interaction. Eindhoven, The Netherlands, Eindhoven University of Technology.

iptel.org (2007). Sip Express Router (SER).

Johnsen, E. G. and W. R. Corliss (1971). Human Factors Applications in Teleoperator Design and Operation.

Karsenty, A. (1997). Easing interaction through user-awareness. Proceedings of the 2nd international conference on Intelligent user interfaces. Orlando, Florida, United States, ACM Press.

Lasswell, H. D. (1948). " The Structure and Function of Communication in Society." New York: Institute for Religious and Social Studies: 203-243.

Ljungberg, F. (1996). An initial exploration of communication overflow. The 2nd International Conference on the Design of Cooperative Systems. Sophia Antipolis, France.

Ljungberg, F. and C. Sørensen (1998). Are You "Pulling the Plug" or "Pushing Up the Daisies"? Thirty-First Hawaii International Conference on System Sciences (HICSS-31). Hawaii, USA, IEEE.

Lombard, M. and T. Ditton (1997). At the Heart of It All: The Concept of Presence. Department of Broadcasting, Telecommunications, & Mass Media, Temple University.

Love, S. (2005). Understanding Mobile Human-Computer Interaction, Elsevier.

Minsky, M. (1980). "Telepresence." Omnia: 45-51.

Norwegian Post and Telecommunications Authority (2007). Det norske ekomarkedet 2006.

Olsen, Ø. (1999). Mobility and Communication - Why context matters. Department of Informatics. Oslo, University of Oslo. **Master**.

PitneyBowes (1998). "Workplace communications in the 21st century."

Rosenberg, J., H. Schulzrinne, et al. (2006). "RPID: Rich Presence Extensions to the Presence Information Data Format." from <http://www.ietf.org/rfc/rfc4480.txt>.

Schmidt, A., M. Beigl, et al. (1999). "There is more to context than location." Computers and Graphics **23**(6): 893-901.

Shannon, C. and W. Weaver (1963). "The Mathematical Theory of Communication." Bell System Technical Journal.

Thanh, D. v. (1997). Mobility as an Open Distributed Processing Transparency. Department of Informatics. Oslo, University of Oslo. **Dr.scient.**

## Appendix A: SER Configuration

```
debug=3      # debug level (cmd line: -ddddddddd)
#memdbg=100
#fork=yes
#log_stderr=no  # (cmd line: -E)
#memlog=5 # memory debug log level

check_via=no    # (cmd. line: -v)
dns=no         # (cmd. line: -r)
rev_dns=no     # (cmd. line: -R)
port=5090
children=2
alias="test-domain.com"

#user=ser
#group=ser
#open_fd_limit=1024 # sets the open file descriptors limit
mhomed=yes # usefull for multihomed hosts

#disable_tcp=yes
tcp_accept_aliases=yes # accepts the tcp alias via option
#tcp_poll_method="sigio_rt"

tcp_send_timeout=1
tcp_children=32
tcp_connect_timeout=1
tcp_connection_lifetime=600
tcp_max_connections=50000

# ----- module loading -----

# Uncomment this if you want to use SQL database
loadmodule "/usr/local/lib/ser/modules/xcap.so"
loadmodule "/usr/local/lib/ser/modules/sl.so"
loadmodule "/usr/local/lib/ser/modules/avp.so"
loadmodule "/usr/local/lib/ser/modules/avpops.so"
loadmodule "/usr/local/lib/ser/modules/tm.so"
loadmodule "/usr/local/lib/ser/modules/rr.so"
loadmodule "/usr/local/lib/ser/modules/maxfwd.so"
loadmodule "/usr/local/lib/ser/modules/usrloc.so"
loadmodule "/usr/local/lib/ser/modules/registrar.so"
loadmodule "/usr/local/lib/ser/modules/textops.so"
loadmodule "/usr/local/lib/ser/modules/mysql.so"
loadmodule "/usr/local/lib/ser/modules/dialog.so"
```

```
loadmodule "/usr/local/lib/ser/modules/rls.so"
loadmodule "/usr/local/lib/ser/modules/pa.so"
loadmodule "/usr/local/lib/ser/modules/presence_b2b.so"
loadmodule "/usr/local/lib/ser/modules/uri.so"
loadmodule "/usr/local/lib/ser/modules/uri_db.so"
loadmodule "/usr/local/lib/ser/modules/domain.so"
loadmodule "/usr/local/lib/ser/modules/fifo.so"
loadmodule "/usr/local/lib/ser/modules/xmlrpc.so"
loadmodule "/usr/local/lib/ser/modules/xlog.so"
#loadmodule "/usr/local/lib/ser/modules/unixsock.so"

# binrpc
loadmodule "/usr/local/lib/ser/modules/ctl.so"

# Uncomment this if you want digest authentication
# mysql.so must be loaded !
loadmodule "/usr/local/lib/ser/modules/auth.so"
loadmodule "/usr/local/lib/ser/modules/auth_db.so"
loadmodule "/usr/local/lib/ser/modules/msilo.so"

# ----- setting module-specific parameters -----

# modparam("msilo","registrar","sip:registrar@test-domain.com")
modparam("msilo","use_contact",0)
modparam("msilo","expire_time",7200)

# -- usrloc params --

# -- auth params --
# Uncomment if you are using auth module
#
modparam("auth_db", "calculate_ha1", yes)
#
# If you set "calculate_ha1" parameter to yes,
# uncomment also the following parameter)
#
modparam("auth_db", "password_column", "password")

# -- rr params --
# add value to ;lr param to make some broken UAs happy
modparam("rr", "enable_full_lr", 1)

modparam("rls", "min_expiration", 300)
modparam("rls", "max_expiration", 300)
modparam("rls", "default_expiration", 300)
modparam("rls", "expiration_timer_period", 30)
modparam("rls", "auth", "none")
modparam("rls", "reduce_xcap_needs", 1)
modparam("rls", "db_mode", 1)
modparam("rls", "timer_interval", 10)
modparam("rls", "max_notifications_at_once", 100);
modparam("rls", "max_list_nesting_level", 4);

modparam("pa", "use_db", 1)
# allow storing authorization requests for offline users to db
modparam("pa", "use_offline_winfo", 1)
# how often try to remove old stored authorization requests
modparam("pa", "offline_winfo_timer", 600)
# how long stored authorization requests live
modparam("pa", "offline_winfo_expiration", 600)
```



```

# mode of PA authorization: none, implicit or xcap
modparam("pa", "auth", "none")
# do not authorize watcherinfo subscriptions
modparam("pa", "winfo_auth", "none")
# use only published information if set to 0
modparam("pa", "use_callbacks", 1)
# don't accept internal subscriptions from RLS, ...
modparam("pa", "accept_internal_subscriptions", 0)
# maximum value of Expires for subscriptions
modparam("pa", "max_subscription_expiration", 300)
# maximum value of Expires for publications
modparam("pa", "max_publish_expiration", 300)
# how often test if something changes and send NOTIFY
modparam("pa", "timer_interval", 1)
modparam("pa", "async_auth_queries", 0)
modparam("pa", "auth_rules_refresh_time", 60)
modparam("pa", "max_auth_requests_per_tick", 1000)
modparam("pa", "ignore_408_on_notify", 1)
#modparam("pa", "pres_rules_file", "presence-rules.xml")
#experimental:
#modparam("pa", "subscribe_to_users", 1);
#modparam("pa", "pa_subscription_uri", "sip:presence-server@test-domain.com");

# route for generated SUBSCRIBE requests for presence
#modparam("presence_b2b", "presence_route", "< sip:127.0.0.1;transport=tcp;lr>")
modparam("presence_b2b", "presence_outbound_proxy", "sip:127.0.0.1;transport=tcp")
#modparam("presence_b2b", "presence_outbound_proxy", "sip:127.0.0.1")
# waiting time from error to new attempt about SUBSCRIBE
modparam("presence_b2b", "on_error_retry_time", 60)
# how long wait for NOTIFY with Subscription-Status=terminated after unsubscribe
modparam("presence_b2b", "wait_for_term_notify", 33)
# how long before expiration send renewal SUBSCRIBE request
modparam("presence_b2b", "resubscribe_delta", 30)
# minimal time to send renewal SUBSCRIBE request from receiving previous response
modparam("presence_b2b", "min_resubscribe_time", 60)
# default expiration timeout
modparam("presence_b2b", "default_expiration", 3600)
# process internal subscriptions to presence events
modparam("presence_b2b", "handle_presence_subscriptions", 1)
#additional headers for presence
#modparam("presence_b2b", "additional_presence_headers", "P-Generated: yes\r\nP-Regenerated: no\r\n")
# randomized SUBSCRIBE requests?
modparam("presence_b2b", "max_subscribe_delay", 10)

#modparam("usrloc", "reg_avp_flag", "regavps")
modparam("usrloc", "db_mode", 0)

modparam("domain", "db_mode", 1)
modparam("domain", "load_domain_attrs", 1)
modparam("domain|uri_db|acc|auth_db|usrloc|msilo|rls|pa", "db_url", "mysql://ser:heslo@127.0.0.1:3306/ser")

modparam("fifo", "fifo_file", "/tmp/ser_fifo")

#modparam("xcap", "xcap_root", "http://pulpuk/xcap")
modparam("xcap", "xcap_root", "http://localhost/xcap")

# ----- request routing logic -----

# main routing logic

```

```

avpflags regavps;

route{
    # XML RPC
    if (method == "POST" || method == "GET") {
        dispatch_rpc();
        break;
    }

    # initial sanity checks -- messages with
    # max_forwards==0, or excessively long requests
    if (!mf_process_maxfwd_header("10")) {
        sl_send_reply("483", "Too Many Hops");
        break;
    };
    if (msg:len >= max_len) {
        sl_send_reply("513", "Message too big");
        break;
    };

    # we record-route all messages -- to make sure that
    # subsequent messages will go through our proxy; that's
    # particularly good if upstream and downstream entities
    # use different transport protocol
    if (!method=="REGISTER") record_route();

    # subsequent messages withing a dialog should take the
    # path determined by record-routing
    if (loose_route()) {
        # mark routing logic in request
        append_hf("P-hint: rr-enforced\r\n");
        route(1);
        break;
    };

#    lookup_domain("To");
#    lookup_user("To");
#
#    xlog("L_ERR", "Dispatch request %rm to: %tu from: %fu\n");
#    ds_select_new("1", "3"); /* request uri */
#    sl_send_reply("302", "Moved temporarily");
#    break;

    if (!lookup_domain("$td", "@to.uri.host")) {
        xlog("L_ERR", "Unknown domain to: %tu from: %fu\n");
        route(1);
        break;
    }

#    xlog("L_INFO", "xcap_root: %t.xcap_root\n");

    if (method=="SUBSCRIBE") {
#        if (@msg.supported=~"eventlist") {
#            xlog("L_ERR", "!!! Support for event lists: %@msg.supported\n");
#        }
#        else {
#            xlog("L_ERR", "!!! NON-Support for event lists: %@msg.supported\n");
#        }
    }
}

```

```

if (search("^(From|f):.*sip:presence-server@test-domain")) {
    log(1,"subscription from PA!\n");
    # subscriptions from PA to user !!!
    if (!lookup("location")) {
        sl_send_reply("404", "Not Found");
        break;
    };
#    append_hf("P-hint: usrloc applied\r\n");
    route(1);
    drop;
};

if (!t_newtran()) {
    sl_reply_error();
    break;
};

if (@to.tag=="") {
    # only for new subscriptions (with empty to tag)

    if (lookup_user("$tu.uid", "@to.uri")) {
        # existing user -> it is subscription to PA

        # xcap parameters
#        set_xcap_root("http://localhost/xcap");
#        set_xcap_filename("pres.xml");
#        xlog("L_INFO", "Hopla!\n");

        $xcap_root = "pokus";
#        set_xcap_root("http://localhost/xcap");
        set_xcap_filename("pre.xml");

        xlog("L_ERR", "XCAP_ROOT before: %$xcap_root\n");
        if (handle_subscription("registrar")) {
            xlog("L_ERR", "XCAP_ROOT after: %$xcap_root\n");
            break;
        }

        if ((@msg.event=~"presence\.winfo")) {
            # new watcher info subscription
            # sends one watcher info NOTIFY message with all
            saved authorization requests

            #xlog("L_ERR", "dumping stored winfo to %fu\n");
            dump_stored_winfo("registrar", "presence");
        }
        else {
            # new presence subscription
            #if          ((@msg.event=~"presence")          &&
            check_subscription_status("pending")) {

                if ((@msg.event=~"presence")) {
                    # if offline user and new pending subscription
                    if (!target_online("registrar")) {
                        #xlog("L_ERR", "storing 'pending'
                        winfo to: %tu, from: %fu\n");

                        store_winfo("registrar");
                    }
                }
            }
        }
    }
}
break;

```

```

    }

    if ((@msg.supported=~"eventlist")) {
        # such user doesn't exist and Supported header field
        # -> probably RLS subscription

        #set_xcap_root("Http://LOCALhost/xcap");

        if (lookup_domain("$fd", "@from.uri.host")) {
            if (lookup_user("$fu.uid", "@from.uri")) {
                if (is_simple_rls_target("$uid-list")) {
                    # if (is_simple_rls_target("contact-list")) {
                    # log(1, "it is simple subscription!\n");
                    # takes From UID and makes XCAP query

                    # list named "default"
                    if (!query_resource_list("default")) {
                        t_reply("404", "No such user list");
                        break;
                    }
                }
                else {
                    if (is_simple_rls_target("contact-list")) {
                        if (!query_resource_list("testing"))
                            t_reply("404", "No such
user contact list");
                        break;
                    }
                }
            }
        }

        if (!have_flat_list()) {
            # query_resource_list failed or was not called
            # do standard RLS query according to To/AOR
            if (!query_rls_services()) {
                log(1, "XCAP query failed\n");
                t_reply("404", "No such list URI");
                break;
            }
        }

        # uncomment this if you want to authenticate first SUBSCRIBE request to
resource list
#
#         if (!proxy_authenticate("test-domain.com", "credentials")) {
#             proxy_challenge("test-domain.com", "0");
#             break;
#         };

        handle_rls_subscription("1");
    }
    else {
        # not resource list subscription -> invalid user
        #xlog("L_ERR", "subscription to invalid user %tu\n");
        t_reply("404", "User not found");
    }

    break;

```

```

    }
    else {
        # renewal subscriptions - try to handle it as RLS and if failed, handle it as PA subscription
        # FIXME: better will be test like existing_rls_subscription()
        #   and existing_subscription("registrar")
        if (!handle_rls_subscription("0")) {
            lookup_user("$tu.uid", "@to.uri"); # needed to get correct UID (internal call
converts it to lowercase!)
            handle_subscription("registrar");
        }
        break;
    }
};

if (method=="NOTIFY") {
    if (search("^(To)t:.*sip:presence-server@test-domain")) {
        log(1,"notify to PA!\n");
        # notification to PA from user !!!
        if (!t_newtran()) {
            log(1, "newtran error\n");
            sl_reply_error();
            break;
        };
        # handle notification sent in internal subscriptions (presence_b2b)
        if (!handle_notify()) {
            t_reply("481", "Unable to handle notification for PA");
        }
        break;
    }
};

# get user (common for all other messages than SUBSCRIBE)
if (!lookup_user("$tu.uid", "@to.uri")) {
    xlog("L_ERR", "Unknown user, To: %tu?");
    # break;
    #append_hf("P-hint: unknown user\r\n");
    sl_send_reply("404", "Unknown user");
    #route(1);
    break;
}

if (method=="PUBLISH") {
    if (!t_newtran()) {
#       log(1, "newtran error\n");
        sl_reply_error();
        break;
    };
    handle_publish("registrar");

    # deliver messages to online user
    # TODO: only if user goes from offline to online?
    if (target_online("registrar")) {
        # log(1, "Dumping stored messages\n");
        # dump stored messages - route it through myself (otherwise routed via DNS!)
        if (m_dump("sip:127.0.0.1:5090")) {
            #xlog("L_ERR", "MSILO: offline messages for %fu dumped\n");
            break;
        }
    }
}
}

```

```
        break;
    };

    if (method=="NOTIFY") {
        if (!t_newtran()) {
            log(1, "newtran error\n");
            sl_reply_error();
            break;
        };
        # handle notification sent in internal subscriptions (presence_b2b)
        if (!handle_notify()) {
            t_reply("481", "Unable to handle notification");
        }
        break;
    };

    if (method=="MESSAGE") {

        if (authorize_message("im-rules.xml")) {

            # use usrloc for delivery
            if (lookup("location")) {

                log(1, "Delivering MESSAGE using usrloc\n");
                t_on_failure("1");
                if (!t_relay()) {
                    sl_reply_error();
                }

                break;
            }
            else {
                # store messages for offline user
                #xlog("L_ERR", "MSILO: storing MESSAGE for %tu\n");

                if (!t_newtran()) {
                    log(1, "newtran error\n");
                    sl_reply_error();
                    break;
                };

                # store only text messages NOT isComposing... !
                if (search("^(Content-Type|c):.*application/im-iscomposing\+xml.*")) {
                    #log(1, "it is only isComposing message - ignored\n");
                    t_reply("202", "Ignored");
                    break;
                }

                if (m_store("0", "sip:127.0.0.1:5090")) {
                    #log(1, "MSILO: offline message stored\n");
                    if (!t_reply("202", "Accepted")) {
                        sl_reply_error();
                    };
                } else {
                    #log(1, "MSILO: error storing offline message\n");
                    if (!t_reply("503", "Service Unavailable")) {
                        sl_reply_error();
                    };
                };
            }
            break;
        }
    }
}
```

```

        }
        break;
    }
    else {
        log(1, "unauthorized message\n");
        sl_reply("403", "Forbidden");
    }
    break;
}

if (method=="REGISTER") {
    # uncomment this if you want to authenticate REGISTER request
    # if (!www_authenticate("test-domain.com", "credentials")) {
    #     www_challenge( "test-domain.com", "0");
    #     break;
    # }

    $t.a = @msg.cseq;
    setavpflag("$t.a", "regavps");
    save("location");

    # dump stored messages - route it through myself (otherwise routed via DNS!)
    if (m_dump("sip:127.0.0.1:5090")) {
        #xlog("L_ERR", "MSILO: offline messages for %fu dumped\n");
        break;
    }
    break;
};
t_on_branch("1");
# native SIP destinations are handled using our USRLOC DB
# check presentity status
$onlinestr = target_online("registrar");
xlog("L_ERR", "PRESENCE - User online: %$onlinestr");
if (!lookup("location")) {
    sl_send_reply("404", "Not Found");
    break;
};
# append_hf("P-hint: usrloc applied\r\n");
route(1);
}

branch_route[1]
{
    #xlog("L_ERR", "on_branch: to: %tu, from: %fu\n");
    #xlog("L_ERR", "ruri: %ru uid: %$.uid\n");
    read_reg_avps("location", "$t.uid");
    #xlog("L_ERR", "$t.a = %$.a");
}
route[1]
{
    # send it out now; use stateful forwarding as it works reliably
    # even for UDP2TCP

    if (!t_relay()) {
        sl_reply_error();
    };
}

failure_route[1] {
    # forwarding failed -- check if the request was a MESSAGE

```

```
if (!method=="MESSAGE") { break; };
#log(1, "MSILO: MESSAGE forward failed - storing it\n");

# we have changed the R-URI with the contact address, ignore it now
if (m_store("0", "")) {
    t_reply("202", "Accepted");
} else {
    #log(1, "MSILO: offline message NOT stored\n");
    t_reply("503", "Service Unavailable");
};
}
```