Universität Karlsruhe (TH)
Forschungsuniversität · gegründet 1825

Fakultät für Informatik

Interactive Systems Laboratories (ITI)

Communication Shields

—

Design and Prototype Implementation of a Context-Aware Communication Service

Studienarbeit by
Daniel Pathmaperuma
perumer@the-magic-team.de

Advisors:
Dipl.-Inform. Maria Danninger
Dr. Rainer Stiefelhagen
Prof. Alex Waibel

July 31st, 2007

I hereby confirm that I wrote
this work on my own and without
foreign help.
I used none but the stated literature.

Karlsruhe, July 31st, 2007

Daniel Pathmaperuma

Contents

1 Introduction 1

2 Related Work 3
2.1 Science . . . 3
2.2 Industrial . . . 6

3 System Design 7
3.1 Caller – the sender . . . 7
3.2 Callee – the receiver . . . 8
3.3 Context . . . 9
3.4 Adjusting Context Awareness . . . 11
3.5 Interface . . . 11
3.6 Data Sources . . . 12

4 Implementation of a Prototype System 15
4.1 Scenario . . . 16
4.2 Model: Interruption Shields . . . 16
4.3 Shield GUI . . . 20
4.4 Sensor Simulator . . . 21
4.5 Setting Preferences & Editing Contacts . . . 22
4.6 Overall System Architekture . . . 24

5 Survey 25
5.1 Objectives . . . 25
5.2 Participants . . . 25
5.3 Questions . . . 26
5.4 Results . . . 28
5.5 Discussion . . . 30
5.6 Conclusion . . . 32

6 Further Research 33
6.1 Future Work . . . 33
6.2 General/Further Ideas . . . 34

7 Summary/Conclusion **39**

Abstract

The world is growing together more and more. Globalization is just one word for this phenomenon. This was made possible last, but not least by the possibilities of modern communication, e.g. mobile telephony, e-mail and instant messaging. But the more omnipresent these new means of communication get, the more they are becoming a burden, as users are constantly challenged with communication requests no matter if they are free or busy and in need of some undisturbed time to concentrate on their work.

This work discusses the necessity and possibility of a context aware system that filters incomming communication requests and by doing so give the user some relief in form of undisturbed time.

Die Welt wächst immer mehr zusammen. Globalisierung ist nur ein Wort für dieses Phänomen. Dies wurde nicht zuletzt durch die modernen Kommunikationsmittel, wie mobile Telephonie, E-Mail und Instant Messaging ermöglicht. Aber je omnipresenter diese Kommunikationsmittel werden, umso mehr werden sie auch zur Belastung für ihre Benutzer, da sie ständig durch Kommunikationsanfragen gefordert werden, egal ob sie gerade unbeschäftigt sind, oder beschäftigt und sich auf ihre Arbeit konzentrieren müssen.

Die vorliegende Arbeit diskutiert die Notwendigkeit und Möglichkeiten eines Kontext-Sensitiven Systems, das eingehende Kommunikationsanfragen filtert und so dem Benutzer etwas Erleichterung in Form von ungestörter Zeit verschafft.

Chapter 1

Introduction

The real voyage of discovery consists not in seeking new landscapes, but in having new eyes

Marcel Proust

In today's world, more and more people do have more and more communication devices. What was meant to be a blessing, nowadays has sometimes even become annoying. Here is an example: Sam wants to ask a simple question to Bob. He calls him, but Bob is not at home. His wife answers and tells Sam that he is in the office. So Sam calls the office. There he just reaches the answering machine, telling him that Bob is not at his desk at present. Sam remembers Bobs cellphone number and calls there. At that time Bob is having a presentation for an important customer. As he forgot to switch of his mobile phone, it rings while he is speaking to his customer. Now Sam and Bob are both in a snit. Sam because he is having a hard time trying to reach Bob and Bob because he got disturbed in his important meeting. Both of them wish things were easier and more intelligent.

This is where the *Connector* idea comes from. Its main goal is to ease communication between people, acting very much like a digital secretary. Connector is a framework of services that offers every user a *personal agent*, that manages their owner's communication needs. In the above-mentioned example, all Sam wanted to do was to get a piece of information from Bob. Instead of calling Bobs multiple phone numbers, he could have simply asked his Connector agent to find a convenient time to contact Bob for the desired information and inform him once Bob is free to talk. On the other hand, Bob's Connector agent could have known that he was in a meeting at the moment of the call and that it would be inappropriate if his phone rang. If Sam's question was important and urgent, connector could have informed Bob discreetly, otherwise he could have waited until after the meeting to connect Sam and Bob. As soon as Bob has time to talk on the phone, he would have contacted Sam's Connector agent that in turn would have informed Sam about the possibility to call now.

This is one aspect of the Connector idea. To achieve this goal there are many sub-problems to be solved, mainly the questions "How does the Connector agent know whether he can interrupt its master right now with a phone call or whether it would be better to use an alternative means of communication?" and "How can the Connector agent decide whether a call is important enough to allowe to disturb the callee?".

As there has been quite some advances in answering the first question, the main goal of this work is to find answers to the second question.

The remainder of this theses is organized as follows. Chapter 2 takes a brief look at the current state of research in related fields. Chapter 3 explains some considerations regarding the design of a system that could be able to make decissions related to the disturb-or-not problem. In chapter 4 the implementation of a prototype system is explained. Chapter 5 discusses a survey relating this topic. Chapter 6 concludes the developement with some ideas for future improvements. Finally some conclusions are drawn that summarizes this work in chapter 7.

In this paper a fictionl user used for the examples is always adressed to as "he". Of course this is not done to discriminate anyone but due to the fact that the author sees himself in these examples and naturally adresses himself as "he". None of the given examples is ment for male users exclusively, any female reader should feel free to replace all male pronouns with female ones in her head while reading.

Chapter 2

Related Work

When a distinguished but elderly scientist states that something is possible, he is almost certainly right.
When he states that something is impossible, he is very probably wrong.

Arthur C. Clarke
— *Profiles of The Future*, 1961

There are several approaches towards this topic. A lot of academic institutions work in this field for a long time now. But in the last decade the communication industry recognized the importance of this topic. Therefore they also started to work in this field.

Naturally the goal of these two groups are slighly different. While the scientific approach aims on increasing the knowledge and understanding of the topic, the industry first of all is interested in working solutions they can sell. Therefore this chapter is divided into related work in *science* and the *industry*.

2.1 Science

Already in 1991, M. Weiser described in [1] the idea of ubiquitous computing. He predicts the use of several hundrets of micro computer devices per room that are intuitively to operate so the user can at last stop concentrating on computer devices and start to care about the people at the other ends of the network. Today, 15 years later, we are still far from reaching his ambitious and undeniably most desirable objective.

Satyanarayanan states in [2] the four essential requirements to *pervasive computing*. The first is *effective use of smart space*, meaning the inclusion of technology into everyday-life and thereby enabling interactive sensing and controlling.

The second requirement is *invisibility*. The final goal is the complete disappearance of pervasive computing technology from a users's consciousness. This requirement automatically leads to the demand of *minimal user disturbance*.

The third requirement is *localised scalability*. Satyanarayanan states that the current ethos of the internet ("death of distance") is inappropriate for pervasive computing as the user as well as his computing system will be overwhelmed by distant interactions that are of little relevance if the density of interaction does not fall of as one moves away. One example for this would be the smart house. It adjusts the lighting condition to the user's mood. This of course is not necessary while the user is not at home, still querying his mood is one kind of distant interaction without relevance.

The last requirement is *masking uneven conditioning*. Satyanarayanan preceives that the rate of penetration of pervasive computing technology into the infrastructure will vary considerably. As this results in huge differences in the "smartness" of different environments, a pervasive computing system will have to compensate these differences in order to keep the user clear of annoyances.

Furthermore Satyanarayanan states that in order to achieve *minimal user disturbance* a pervasive computer system must be *context-aware* so that it can adjust its behaviour to the user's currrent state and surroundings. A very important issue in this context is the balance between *proactivity* and *transparency* as the user's needs depend on his level of experience on a task and his familiarity with his environment. A system can easily annoy a user be proactively initialising all kinds of processes as well as confuse him by masking activities behind undesired transparency.

Mummert et al. present an interesting approach to this problem. In [3] they mention a *user patience model*. In their experiment the users had mobile devices with limited memory but the ability to get unavailable data through a low bandwith-connection. The patience model was used to decide whether a user should be queried what to do on a cache-miss on the mobile system. They assume, that a user is more likely willing to accept a disturbance if the missing file is of a higher importance. As there is no scientific data available by now, they conjecture a logarithmic relation between patience and importance, as they claim that many other human processes have a similar sensitivity (e.g. vision).

In [4] Coutaz et al. extend the notion of context not only to describe a state but a process. They argue that only a *context-as-process* approach is flexible enough to avoid mismatches between the system's model of interaction and the users's mental model of the system.

In [5] Dabbish et al. examine how people react on incoming mails. They classified the incoming message content to one of the following classes: *request for action, request for information (link, document, contact info etc), status update (for ongoing project/task), request for a meeting or response to a meeting request, reminder or social greeting*. Also they took into account the relation of the sender to the recipient and the importance of the message (to the recipient as well as to the sender). Their goal was to derive a model that allows to predict the user's reaction to an incoming mail. Contrary to their expectations, it was not the message importance that had the highest influence on the reply-behaviour of the users but the *sender characteristics*. The authors also noted that messages requesting information were significantly more likely to receive a response and assume that this is because the appropriate action on these messages is most apparent to the recipient. Another interesting side note is the fact that messages with social content were more likely to receive a response.

In [6] Erickson et al. argue, that current technology is creating walls between users. The suggested approach is to make systems *socially translucent* [1] to enable the users to use their common social behaviour to communicate more efficiently with one another.

In [7] Olson et al. express their opinion that no matter how sophisticated a communication or collaboration system might be, distance will always play an important role, as there are to many facets in human-to-human communication that technology can hardly or not at all reflect.

In addition, M. Schneider et al. observed that not only physical distance but more important *social network distance* plays a significant role on the efficiency of human to human communication [8]. Hence they claim that human-computer interaction as well as computer supported cooperative work could benefit by increasing focus on social network distance as an analytical measure and design tool.

In their surveys [9], J. M. Hudson et al. found that availability management can not be just evading all interruptions as there is a tension between wanting to avoid interruptions and appreciating their usefulness. For example in their study Managers spend most of their worktime handling interruptions and regard this part of their job. They need a certain control over their interruptions to keep sense of their projects on the one hand but on the other hand also need some time of *deep concentration*. During these times it is important to keep any interruption away, as the temptation to let themselves be interrupted in order to not delay any potentially important information is to strong for most of them. These times of deep concentration arise quite regularly in a workday. In summary Hudson et al. demand systems to make interruptions more efficient rather than decreasing them. As they also found that interruptions are handled differently depending on the nature of the interruption regardless of state or time, they believe that *socially translucent systems* [6] are the best way to approach this challenge.

Another important finding of them is that systems that rely on users updating their current status manually are bound to be ineffective as users of these systems never seem to remember to change their availability state.

Forgarty et al. examined this opinion [10] and found that people tend to ignore the availability information provided by their *MyVine* social translucient availability information software. (MyVine uses speech detection, location information, computer activity and calendar information to model a user's availability.) Contrary, they use the provided information to determine the *presence* of a person rather than its *availability*. Hence the auhors implicate that a socially translucent system is a good approach but far from enough to alleviate inappropriate disturbances significantly or prevent them totally.

[1] socially translucent means that the communication channel transmits not only the content of the communication but also the emotions of the users

2.2 Industrial

There are a couple of groups that are working on the future of (mobile) communication. Two of them are the 3rd Generation Partnership Project [11] and the Open Mobile Alliance [12] [2]. One of their projects is called *Presence Service* and specifies a service that enables the user to define certain triggers that are connected with his current status. On this level it is not specified how the current status is changed, what states there are at all or what kind of actions could be triggert. The goal of these working groups is simply do specify a common architectual basis on which the single members could implement their own solutions and devices. The common basis guarantees the interoperability of all solutions and devices within the specifications.

The Finnish communication technology company Nokia [3] developed a technology called *Nokia Presence* [13] that seems to be based on the *Presence Service* definded by the 3GPP. It consists of the*Presence Server* and the *Presence Client Software* and allows users to exchange reachability information. A user can set his reachability status (busy, away, bored – please call) as well as his preferred mode of communication (voice call, SMS) along with some personal notes. A status change can be triggered either manually by the user or automatically by a calendar event. This information is stored on the Nokia Presence Server where other users can query them. Presence offers a group based privacy concept that allows the users to decide who can get what kind of information about them.

Nokia lists over 30 cell-phone models that already support Presence. The Presence client software is available for download on the Nokia website and it is said that it also works with phones manufactured by other companies. Although it communicates via GPRS, EDGE or WCDMA it depends on some provider depending network services (mainly the presence server) which are currently not supported by any German network provider company.

[2] please note that these two groups are overlapping, members of the 3GPP are often also members of the Open Mobile Alliance and vice versa

[3] Nokia is possibly and probably not the only company that already has products that implement the *Presence* technology, but it seems to be the only company that mentiones it in its advertisment campaigns. This is why this company was chosen to be an example here.

Chapter 3

System Design

The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.

Mark Weiser
— *The Computer for the 21st Century*, 1991

Given all the above mentioned, it seems wise to collect all expectations towards a smart communication system before starting to design it. As a communication system connects (at least) two people, there are also at least two interests to be attendet to. *(Please note the the terms* caller *and* callee, *that are derived from the telephone context, will be used for the initiator and the receiver of* any *form of communication, telephone as well as mail or instant messaging.)*

3.1 Caller – the sender

The first faction is the caller side. The caller has a certain motivation to initiate the communication. These motivations can be classified into three major groups. While the first two of them are examples for asynchronous communication as the partners do not have to interact in realtime, the third one obviously is an example for synchonous communication.

The first is the impartation of information, which means tha the caller mainly wants to inform the callee about something. This information is fixed and can be delivered without any involvement of the caller. Typical examples of imparting information would be messages like "I'm out of office", "Task XY is done" or "The meeting was rescheduled to 15:00h".

The second is the request of information. As like as the impartation of information the request can be delivered without any involvement of the caller. Typical examples of requesting information would be messages like "What is the status of project XY?", "Please send me the document containing XYZ", or "What is the name of the new secretary again?".

The third is the desire to interact with the callee. This may be to discuss something with the callee or simply to have a nice conversation.

It is obvious that in the first two cases, the actions could be handled by a commuication system without the callee having to wait for the accomplishment, e.g. the caller could order the system to inform the callee about a certain matter and then forget about it. The system would then try to reach the callee until he or she has the time to answer the call. The same could be done with an information request. The caller would simply order the system to "Get the document containing XYZ from user ABC and report back to me, once you have it." Now the system can contact the user ABC, tell him about the desired document and report back to the caller, once the document is available. In the meantime the caller could care about other problems.

Unlike the first two cases, which can be handled asynchronously, the third case requires both parties (caller and callee) to be available at the same time. While a direct system support is hardly possible in this case, a system could still support the users e.g. by finding a point in time where both are available and schedule the call there.

In general it can be said that synchronous communication is troublesome for the caller as he does not have any information about the callee's situation.

3.2 Callee – the receiver

A callee on the other hand has different requirements towards a communication system. He wants to be available for incoming calls to get interesting and/or important information and news. Still, while he is busy, he does not want to be disturbed as well as while being in an situation where a call would be inapropriate. E.g. a stock broker has to be informed about any important event concerning his investments. If he is not reachable at his desk phone, he would want the system to know that in this case he is most likely available on his mobile phone. Still it is highly arguable if this information should come to him while he is attending divine service. An analogue situation can be constructed for the project manager. While any delayed information flow could lead to a delay of the whole project, the manager might still not want to be disturbed while in an important milestone-meeting with a customer. Even if he is just working for himself, an unrelated disturbance might break his concentration, so he would want to be disturbed only by relevant information.

While most of the features mentioned above can be achieved with todays common technology, this would require a lot of manual system maintenace by the user e.g. programming the routing of calls and remebering to switch the mobile off at the beginnig of a meeting and on again after that meeting. It is highly desirable for a communication system to *just know* when a user is available for disturbances and which calls are relevant to his current task.

Another problem for the callee is the distribution of his phone numbers. Nowadays people have many numbers, e.g. office phone, mobile phone, home phone and so forth. Now not everyone wants to share all his numbers with everyone. There are allways people that e.g. should have access only to the office number.

3.3 Context

To decide what to do with a pending call, the communication system must know as many things about the context of the callee as possible and idealy about the context of the caller, too.

This information includes the current location of the callee (office desk, car, home, ...), his current affiliation (project X, customer Y, ...), the situation he is in (alone, conversation, meeting, church service, ...) and his environment (train station, library, ...) along with the available means of communication in this context as well the importance, topic, and urgency of an incoming communication request.

Based on these facts, a communication system would have to decide if a call should be routed, delayed, rejected or handled in another way. While such a task is already hard under laboratory conditions, a computer system would most likely fail in real-life-situation at the lack of needed information because of their complexity.

E.g. it might be possible to identify a person using a bunch of cameras together with a powerfull image-processing-algorithm but it is illusive to expect these camera-systems to be widespread and available for public use at any point in the near future.

Therefore it is necessary to develope cheap and simple sensor components that are easy to integrate into todays communication systems or maybe even *already are* integrated into these. These simple sensor components and the algorithms that combine a lot of simple sensor data to complex context descriptions are a basic requirement for smart communication systems.

Sensor Components

There are many examples for those simple sensors providing high-value information. Some examples will illustrate the idea in brief. For more details, please see Section 3.6.

A system was developed in [14] that was capable of distiguishing eleven different environment classes by their ambient noises. In [15] this system was developed further to run on a smartphone.

Almost every modern PC operating system contains a keyboard (respectively mouse) activity scanner. It enables the PC to determine whether the user is currently active on his PC or if he is away. In the latter case the screensaver could be switched on and the instant messenger set to the away-mode.

Another rich source of data is the calendar. Many companies run e.g. an *Outlook Exchange Server* for their employees. All kinds of information like meetings, conferences or buisiness-trips are registered there. This could enable a system to guess what kind of task a user is currently working on, if he is available for calls or currently in a meeting and when the probability to reach someone at his desk phone is highest.

One last example is a bluetooth proximity scanner. It scans the vicinity for other bluetooth-devices and is able to recognise them and thus recognise their owner. That way a system could know how many people and even who exactely is around at the moment. An implementation of such a scanner is *Nokia Sensor* [16]. Nokia developed a software that lets bluetooth-enabled phones listen for other phones that run *Sensor* in the neighbourhood and react to them. Nokia currently lists 8 compatible phones.

In [17] a mobile phone is combined with some additional sensor components, such as voice- and ambient-noise-microphones, motion-detectors, and light- and temperature seensors. These sensor data is then used to guess the user's interruptability.

All sensor components mentioned above can be described as *software-sensors* or *soft-sensors* as they use existing hardware and convert them to sensor components just by adding the necessary software smartness.

A different type of sensor component is a face recognizer or a meeting detector in a smartroom. These sensor components typically consist of a number of cameras together with some image processing algorithms. They require a huge effort in installation and adjustment and are not available everywhere, furthermore they are not mobile.

Combining Sensor Data

While a smart communication system needs to have a pretty exact picture of a user's current context, the above mentioned soft-sensors provide only very limited information for themselves.

Nevertheless, this data can be used to create a complex description of the user's current state. A calendar entry like "project-meeting" in combination with typical conversation noises and the bluetooth cell-phones of certain colleagues could indicate that the user is attending a meeting. While this data merging is a simple example of what could be made out of basic data, even more is possible.

By logging all sensor data with timestamps, one gets an *event history*. Exploiting this, it is possible to guess the user's context more precisely than without it. This shall be demonstrate by the following short example.

The sourround noise detector is detecting "trainride", no entry is found in the calendar for now and no known persons can be detected in the vicinity. Now, all we could tell by simple data merging is that the user is currently riding on a train. Now we will have a look at the *event history*. Reconstructing the last 30 minutes of the sourround noise sensor, we find: eating (13 min), stairway (1 min), walking on the street (10 min), riding train (6 min). Looking through our event history, we find simmilar patterns in the morning of every working day. Now we know *where* the user is, not just *riding a train* but *riding train line 4 since 6 minutes, probably* *leaving it in another 2 minutes* and we know what the user will probably do next, like *walking for another 8 minutes and reaching his office at 9:03h.*

While the last information is pretty accurate, it was gained by the same basic sensor components mentioned before, only this time the event history was used (along with some data-mining algorithms, of course).

3.4 Adjusting Context Awareness

As most communication systems are designed for humans, it would be nice to have them *behave* like humans, too. By this the acceptance of a system by its users could be increased. One example of a human like behaviour is the decission if a call should be routed to a user while he is coding. Normally a coding user should not be disturbed by calls because this would interrupt his concentration. Now for humans this concentration needs some time to build up. Knowing this a system could route even quite unimportant calls within the first minute of coding, important calls maybe even within the first three minutes. After maybe five minutes of uninterrupted working, a user has reached a level of deep concentration. Disturbing him now would interrupt this concentration.

A similar trade-off could be made for the relevance of interruptions, if the topic of the user's current activity is know as well as the topic of the disturbance. Humans tend to value recent events higher. If someone was talking on the phone about a certain matter only two minutes ago and is having a conversation now, he might still receive a call concerning the matter discussed two minutes ago. This relevance fall-off could be approximated by a hyperbola-like funtction valuing a recent event much higher than one further in the past.

By differentiating between different functions for different situations this human-like behaviour could be improved even further.

Please note that all times and durations mentioned in this section are estimated and not based on actual research findings. Before implementing anything similar, some studies should be made to acquire the real timings e.g. how long it takes for a user to reach a state of deep concentration.

3.5 Interface

One of the most important parts of a system is the user interface. It does not only present the functions of a software to a user but is in fact the *face* of the system. The user recognises and identifies the system by its interface. This is why it is important to have a good interface, not only to make all the system's functions available but to gain acceptance by the users.

Therefore, the interface of an advanced communication system must be simply and clear. No matter how complex the system really is, the user should allways be able to have an idea of *what* a system is doing and *how* it does it. A user might *get* *used* to a system that is a *magical black-box* to him but no user will trust a system unless he has a notion on the systems concept.

As a communication system is never to be a stand alone solution, there are certain further demands to its interface. It must be easy to integrate into existing solutions and means of communications. Idealy it fully integrates itself into existing infastructure so the user will not even know that it exists.

Furthermore, all data used by it must be synchronisable with other systems or software. This has several reasons. First of all, data can be gained quite easily if it is possible to read e.g. address-book or calendar information from other applications. Then there is the trust that a user can have in a system that synchronises with virtually everything. (E.g. it is never a vain endeavour to introduce contact data into *Microsoft Outlook* as almost every system (be it software or hardware) offers an interface to it.)

3.6 Data Sources

Now we shall take a closer look on the relevant context information we know or could know about the user, as well as the ways to get this information.

For now we will ignore all concerns about protection of privacy and the effort that might be necessary to get certain informations, we will concentrate on the *possibilites* these information will give us.

Although we talk about *the system* in the following, a clear definition of this system is still to come. Until then we consider it to be a collection of technologies and data without a certain physical representation.

Location

The information about the whereabout of a user is a very nice thing, if known. If the location is known, one might be able to deduce the current task of a user (or at least get a hint) as well as the possible ways to contact the user.

This information can be gathered in different ways. The most simple one would be a GPS (or Galileo) device that transmits the current position of the user to the system.

Another way is by detecting transmitters, be it a GSM pylon or a Wi-Fi access-point. This is normally not that accurate, but it is better to have a clue than none at all. Also these information can get quiet valuable if paired with e.g. known user habits, user's calendar, etc.

Within a *high surveillance* environment it might be possible to track the user by using cameras, e.g. the current *smartroom evironments* of the CHIL-project [15] are able to track a user moving through the room and identify him on the basis of face recognition.

Environment

A software was developed to run on a smartphone [17]. It listens to the sourrounding sounds, analyses it and clasifies certain *settings* (like a car driving, a crowded pub, walking outside..).

As this application runs directly on the smartphone it is independent from any other data scource and/or technology. The gathered information can be used at once onboard the phone or it might be send to the system. This way it is possible to e.g. adjust the ringtone-volume according to the environment.

Personal Calendars

Electronic calendars might be another rich source of information about the task a user is performing at a certain time. Although the information within a calendar normally is not very detailed, it might help us in deciding what a user is currently doing as well as in determining where he is.

The most important feature of the calendar is the possibility to take future events into consideration, e.g. the return date from the holidays, the end of the meeting or the arrival-time of a plane.

Collocated Persons

In many cases it can be interesting to know which people (if any) are in the close vicinity of the user. This can be especially important to know when the system has to trade off whether the user can be disturbed or not by comparing the importance of persons to the user. E.g. while talking to his superior, it might not be the best time to be interrupted by a buddy who just wants to invite you to a party of golf this afternoon. On the other hand, while discussing the football-results of the weekend in might be perfectly allright if a superrior interrupts to ask about the progress of the current project.

While this information might be easyly available within a *high surveillance* environment (like a smartroom), it is much harder to get in the *real-world*.

One theoretically possible way might be a software similar to the surrounding detection software of the smartphone. It could detect *voices* in the vicinity of the user and then try to relate them to specific persons via voice recognition.

Another possibility might be to try to identify PDAs or phones of people close by via Bluetooth, Wi-Fi or similar wide-spread technologies.

Current Task

The current task is probably the piece of information that is the hardest to get. A current user task could be any out of a huge variety of possible tasks ranging from simple reading or talking on the phone to more complex situations like dealing with a group of business partners in the waiting lounge of an airport.

There is no specific sensor or set of sensors that could detect any out of all tasks of a user. Therefore the *task-detector* has to be a virtual sensor that combines all available information and then derives a (or more likely a possible set of) current task(s).

Chapter 4

Implementation of a Prototype System

Any sufficiently advanced technology is indistinguishable from magic.

Arthur C. Clarke
— *Profiles of The Future*, 1961

As the preceding chapters illustrate, the creation of a semi-intelligent system with satisfatory characteristics is a major task. Still it can be done. As a proof of concept a *light version* of a possible connection manager was implemented. It is called *light version* because by far not all features possible or even mentioned could be implemented. Phone calls received through a VoIP server are processed, checked and possibly routet to the user. Additional information about the caller is requested through a voice interface with recognition of dialed numbers (DTMF).

To keep things simple, no real sensor data is used (although quite some of the required information could have been made available with todays sensor component technology). Instead a *sensor simulator* (see 4.4) is used.

Taken together, all these parts represent a simple connector agent. Incoming calls are routed through a VoIP server using Asterisk [18]. The caller is queried for the importance and urgency of his call using a text-to-speech interface combined with DTMF input. Then the caller is looked up in the address book of the callee. The current context (location, situation) of the callee is determined as well as his personal preferences concerning both his current context and the caller. The decision logic now finds the best way to deal with this call. Finally, the result is displayed on a pop-up window (the shield gui) and the caller gets informed, again via a text to speech interface and the call gets routed respectively blocked. As a result, the phone of the callee will or will not ring.

4.1 Scenario

To give the reader a more vivid impression of this concept, I will issue two examples:

An insurance agent found Bob's number in the phonebook. He calls him to check whether he might be interested in changing his car insurance. As Bob does not know the agent, he is not mentioned in his address-book. Thus the system has no hint on how to handle this request.

Another situation would be the following: Bob is in a meeting with his colleagues where they discuss their current project. While in this meeting, three people try to contact him using the system. The first is his wife. She just wants to say hello but has no urgent matters. The system tells her, that Bob is busy in a meeting right now. She is advised to leave a voice-message, which she does. Shortly after, Sam wants to call Bob, he works at the same company but on a different project. As he is interested in Bobs advise about a certan idea, he has to talk to him before closing time. The system tells him, that Bob is unavailable at the moment but schedules a call for both of them after the meeting. The third person calling is Carl, Bob's superior. He is out of town this week and just got informed that a special client will arrive at the office in a few minutes and someone has to take care of him. He calls Bob using the system and tags this call as urgent and important. The system informs Bob discretely about this pending call, so Bob can get out of the meeting, take the call and thus take care of the client.

This different handling of the request is only possible, because the system knows on the one hand what the user (in this case Bob) is doing currently and on the other hand who is posting the request, how this person is related to the user and how important or urgent this request is. These requests shall be called qualified requests.

Obviously, the *qualified calls* are the interesting ones as only here any detailed processing can be done. This is why the focus lies on processing such calls.

4.2 Model: Interruption Shields

First of all, a basic model was requiered that could fullfill all demands. A *shield* that is build around the user seemd to fit in. It has the benefit of being flexible enough to make it possible to include all kinds of parameters but remains understandable as in the end a strong shield will result in a better protection than a weak one.

Protection Value

The basic idea of the *shield* is that it protects the user from undesired disturbances. Depending on the users current task, it has an initial strength – the *protectionValue*. It is depending on the user's current task, e.g. a task like *coding* results in a stronger shield than *being idle*. Each situation has a prefixed standard value which can be modified by the user to adjust it to his personal preferences with the address book and preference editor.

Topic

Additionaly, the shield gets a *topic*. This topic represents the "project" the user is currently involved with. The topic could be anything, ranging from low-grained information like *work* or *private* to fine-grained information like *client meeting on monday* or *Lisa's english homework*. The topic is used to determine whether the call is related to the user's current task or not.

Caller Importance

Another factors of influence on the shields strength is the *caller importance*. A disturbance by a person that is important to the user is more likely to be accepted than by a person that the user rates as unimportant. For example, if a user is currently working on an important presentation for a client, he might not want to be disturbed by someone asking him about the seat-placement for his daughter's birthday party.

This reveals another aspect of a persons importance: it is often combined with a certain *topic*. Someone might be important for a certain project but totally unimportant to other projects, e.g. while a user is in a syndicate-meeting consulting his partners, his assistent in the project group might have a high importance value while the same person has a low importance value whenever the user is at a clients office talking about future buisiness deals.

Therefore the *caller importance* can be set topic-wise. As (on a scale from 0 to 1) 0.5 is taken as default importance (in cases where no importance is defined for a certain topic), it is possible for the user to *downgrade* certain users in certain situations. This way it becomes possible to assign the lowest possible importance to the annoying insurance broker who just keeps calling for all situations except idle. Accordingly, a high value could be assigned to teammates in certain projects.

Call Importance

Of course the importance of a call can not be assessed by simply looking at the caller's importance to a project. Even an important person might make unimportant calls (your project partner might just want to know if your are free for lunch) or an unimportant person might have something really important (the student apprentice might want to inform you that he messed up the presentation you are about to present). This is why the caller rates the importance of his call seen from his point of view – the *call importance*.

This holds the risc of callers always rating their calls as *very important* but as this can be countered by the user-defined *caller importance* it is worth the risc.

Importance Factor

Combining the *caller importance* and the *call importance* results in the *importance factor*. It represents the actual importance of a call to a user in a certain context. The importance factor is calculated by multiplying the caller importance with the call importance, resulting in values in between one and zero.

Urgency

Finally, a caller is able to state the *urgency* of his call. This allows the system to come up with a good alternative in case a call can not be routed through imediately, e.g. the system could propose a later time to call in case the call is not urgent or it could enumerate alternative means of communication that might reach the user sooner (the user might not be able to take phone calls while attending to a meeting, but might still be able to receive and answer instant messages).

Decision Logic

Now that all parameters necessary to process a call are clear, a decission is needed. The *decission logic* decides whether to route a call to the user, block the call or query the caller for additional information.

First of all, the topic of the call gets compared to the topic of the user's current task. It is assumed, that similar topics indicate a certain relevance of a call while differing topics indicate calls that are of no relevance for the current user's activity.

As an irrelevant call is a potential disturbance, this call should at least be important to get routed. Therefore the importance factor is checked next. If it is larger than the shields protection value, the call will get routed. In the other case the call would be blocked.

But before this is done, the importance of the caller gets checked. If it is larger than the protection value of the shield this means that the caller might have a chance to interrupt the callee but has rated the importance of his call low. In that case he gets a second chance as the system will inform him about the users current context and then ask if the caller would like to reconsider the importance of his call or call back later.

If the caller is not important enough to interrupt in the current context the call is blocked.

Now that it can be assumed that a call actually *is* relevant for the user's current context (as the topics are the same), the urgency of the call gets checked. If the call is urgent, (at the moment a call is considered urgent if it has an urgency value of larger than 0.5) it gets routed to the user emediately.

If it is not urgent the importance factor is checked against the protection value of the shield and in the following the call is handled just like a call with a different topic.

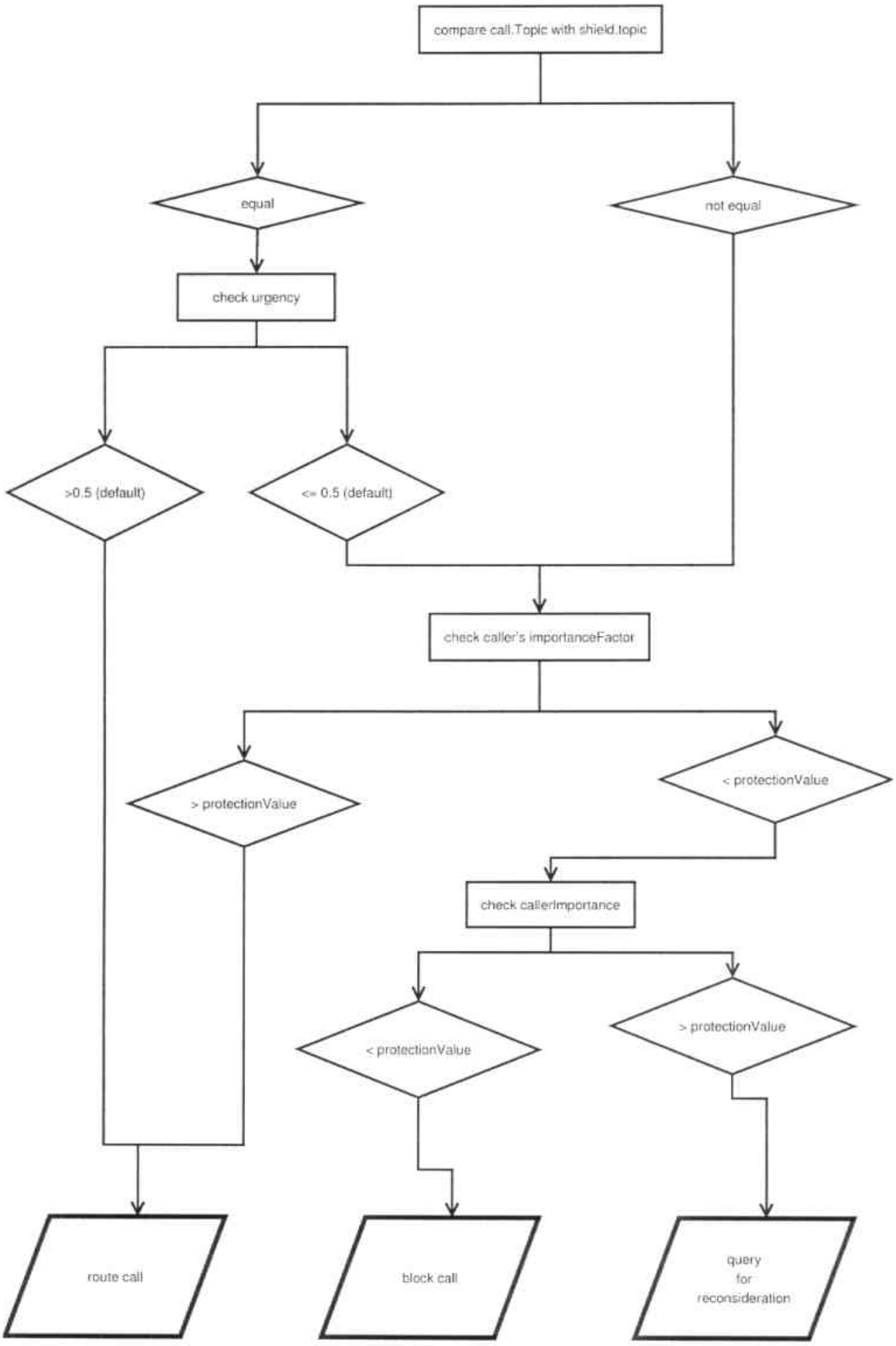

Figure 4.1: the decission tree showing the processing of a call

To outline the system's behaviour one can say that calls that are urgent and concern the same topic the user is currently involved in get routed through emideately. In all other cases the call must be important as well as the caller to get routed. If only the user is important, he is queried for a reconsideration of his call's importance. In all other cases the call gets blocked.

As the user might still wish to avoid any barrier in between him and a certain person calling, it is possible to allow a *system override*. This feature is meant for colleagues or friends who know the user good enough to decide on their own whether they could interrupt the user in a certain context. In case of a potentially inappropriate disturbance, the system would propose a certain action and then ask this *trusted* user if he would like to follow the given proposal or nevertheless continue the call.

Feedback

After processing a call request, a feedback message is presented to the caller. In the case of a rejected call, the feedback should give the caller an idea of what the user is currently doing and why he is unreachable at the moment. As a user might not wish to share this information with everybody, it is possible to control the level of information the feedback provides for each address-book entry depending on the time of day. Possible information contain the *location* and *situation*. The availability of this information can be enabled by the user depedning on the time of day, currently the possibilities are *worktime*, *freetime*, *always* or *never*.

4.3 Shield GUI

As mentioned before, it is vital to the users acceptance that he understands and gets an idea of what the system is doing. Therefore, a pop-up-window will inform the user about any incoming calls and the measures the system performed.

As Plaue et al. found in [19], humans have the ability to get information from an image much faster then from text. Also the adaption of visual data is less interfering than text messages are.

The *Shield GUI* enables the user to see if a call was blocked, routed or passed on for further query in the wink of an eye. For that purpose a large shield icon is presented in the lower left corner of the window. It is green if a call is routed, red for blocked calls and orange for calls that need further query.

Furthermore it shows all the preferences of a call, as well as the decision of the shield logic along with the "reason" for the decision, the reply messages and the alternative communication devices.

Figure 4.2: The Shield GUI showing the result of a processed connection request

4.4 Sensor Simulator

As the integration of real sensors seemed to be a huge effort, a *sensor simulator* was used. With it, the context of a user can be set. In a later state of developement this interface could be replaced by actual sensor data.

As it is, the sensor simulator enables the user to set his current location, situation and topic. The set of possible locations was predefined as *home*, *office* and *car*, as for the situation *reading*, *coding*, *conversation*, *meeting* and *idle* are checkable options. The topic of the current task can be entered as free text.

Of course this limited number of possibilities is not enough for real-life use but it is enough to demonstrate the idea.

The context can be set for each user. On each update, a new entry containing the current timestamp is created in the database. Thus the current context of a user can easily be selected. Keeping older context data makes data mining and machine learning possible at a later date.

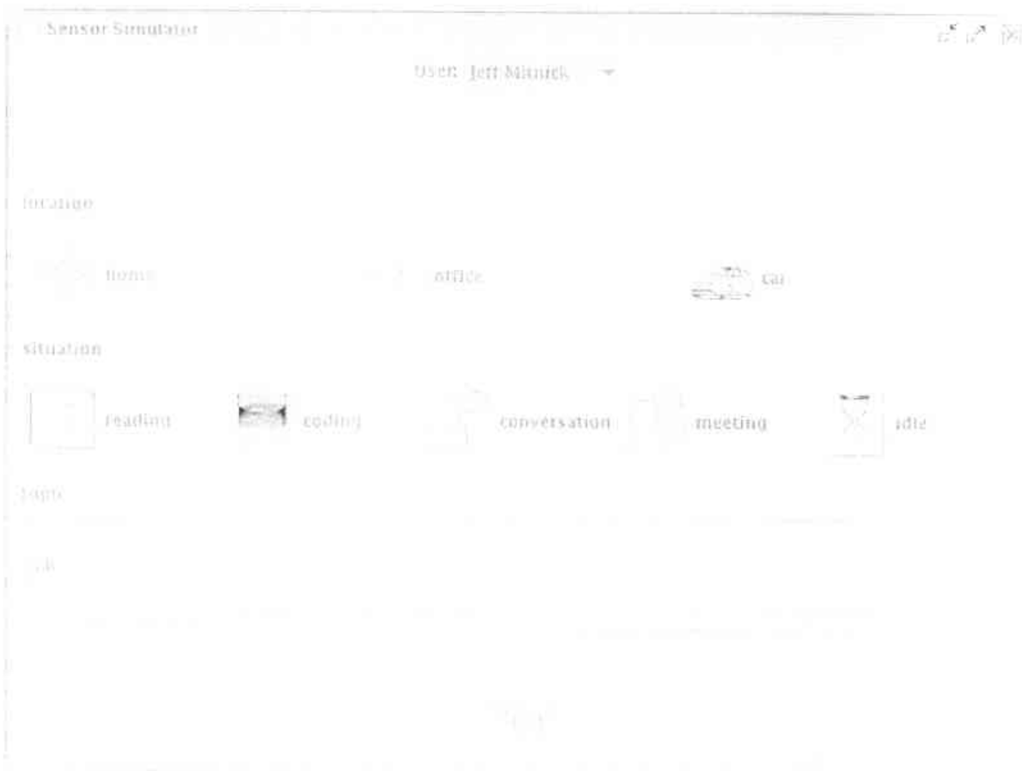

Figure 4.3: the sensor simulator

4.5 Setting Preferences & Editing Contacts

To give the user an easy way to configure "his" connector agent, there is a PHP-based editor, that allows the user to manage his address book including his privacy settings as well as his preferences concerning the shield.

It is necessary to enable the user to use all the above-mentioned features and still keep the system usable without the need to be initialized with hundreds of settings. The system could be initialized with some standard values and is able to function even if some data is not individually set. In such a case the default value is taken. So the user must change the default preferences and settings only if he is not satisfied with the systems reaction. This can be done in an integrated address-book and preference editor.

Contrary to the rest of the implementation the *address book editor* as well as the *preference editor* is a PHP-based dynamical HTML-page. This enables the user to edit all personal settings anytime with a maximum of flexibility. His office PC could do as well as an internet-terminal at the airport. As most modern mobile phones are able to diplay simple HTML-pages even such a device would be sufficient.

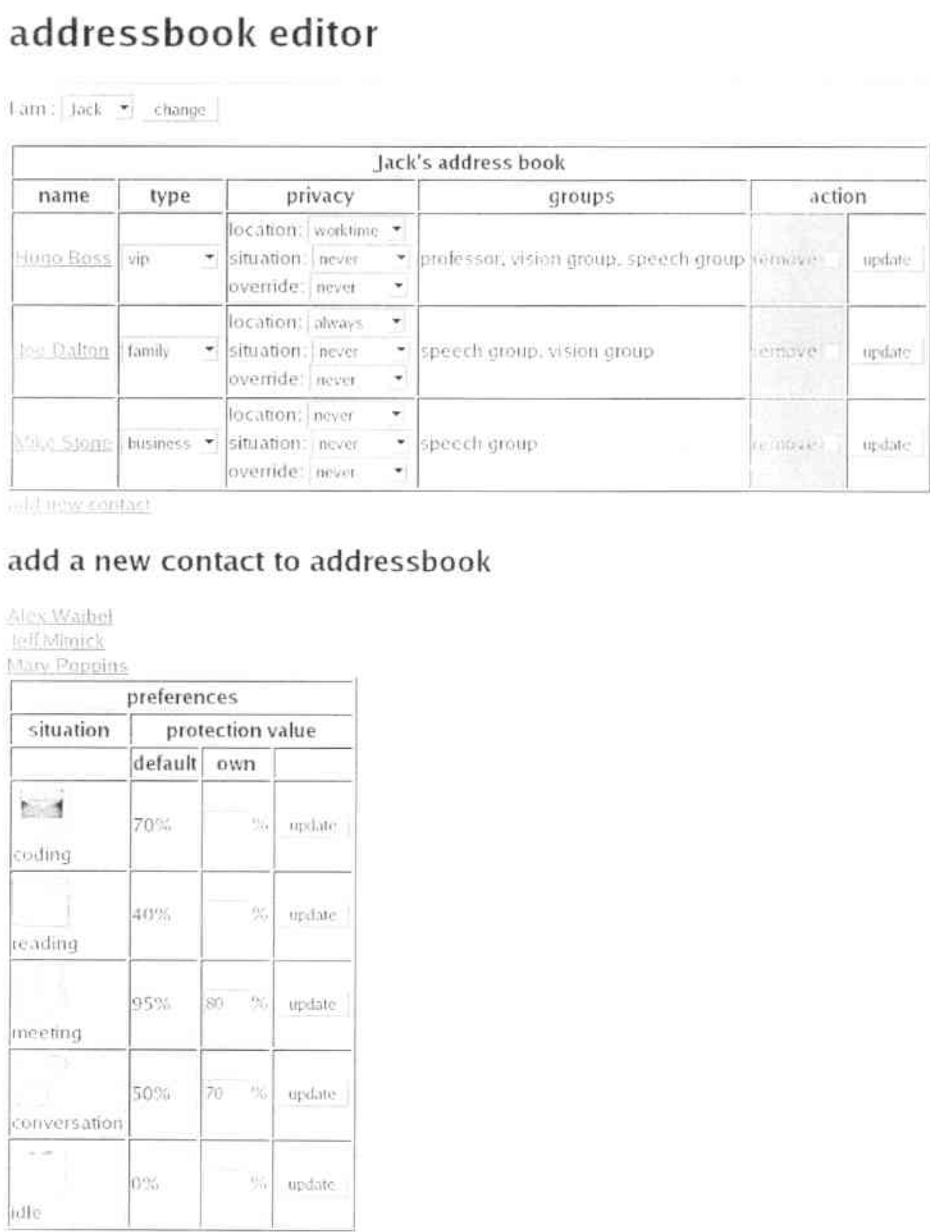

Figure 4.4: the editor

4.6 Overall System Architekture

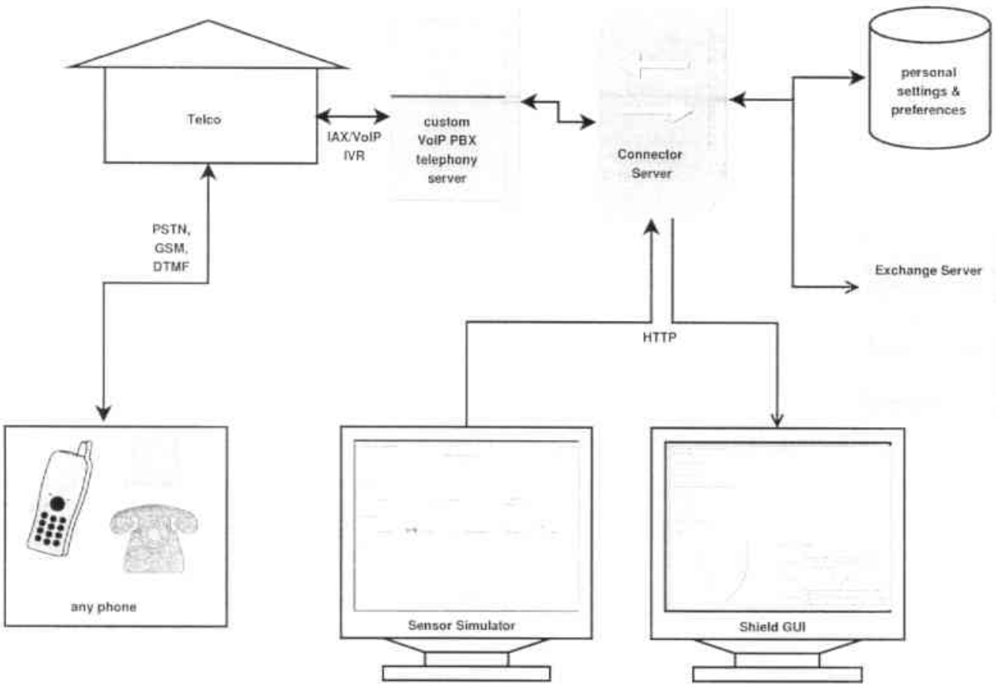

Figure 4.5: Overall System Architekture

A closer look at the *overall system architekture* gives an impression of how all the above described parts fit together.

The number that gets called leads to a VoIP server. It is of no matter if the original call comes from a classical phone (fixed line or mobile) or from a VoIP phone/software. The call then gets routed to the connector server. Here the asterisk server is running.

The user's personal preferences are stored in a database. There they can be accessed by the connector server. They get updated through the preference editor. In the same way the "simulated" sensor data get to the database. There they also can be accessed by the connector server. Also the connector server could access an exchange server to get address book information about the caller.

Now that the connector server has all information it needs to get to a decission it visualizes the call via the shield gui. Now the call gets routed through to the callee or it gets blocked.

Chapter 5

Survey

At the present rate of progress, it is almost impossible to imagine any technical feat that cannot be achieved - if it can be achieved at all - within the next few hundred years.

Arthur C. Clarke, 1983

To get an idea of what potential users think about comunication assistance systems simmilar to the one mentioned in this work, an online survey was conducted. The goal was to collect empirical data about people's attitude towards such a system.

5.1 Objectives

The objective of this survey was to investigate people's attitude towards communication assistance systems.

The main focus was to determine whether people would find such a system useful in their daily life. It is assumed that office phones ring all the time and people get disturbed by it. The survey aimes to verify this.

Another point of interest is whether people would trust a technical system to interfere with their human-human-interaction. Aspects of these questions are the necessity of being observed by the system as well as the need to adjust the system to personal preferences and the effort needed to do so.

Finally it was asked whether they would pay for such a service or if they were in doubt that something like this could ever be implemented.

5.2 Participants

A total of 33 people took part in the survey. Approximately half of them were students, the other half were business professionals. Most of the business professionals classified themselves as working in developement or science. Four classified themselves as working in consulting or management.

Most of the respondents were in between 20 and 35 years of age. Furthermore the vast majority of them can be considered as Germans and working/studying in German companies or universities.

5.3 Questions

All in all there were three groups of questions, ten questions in total.

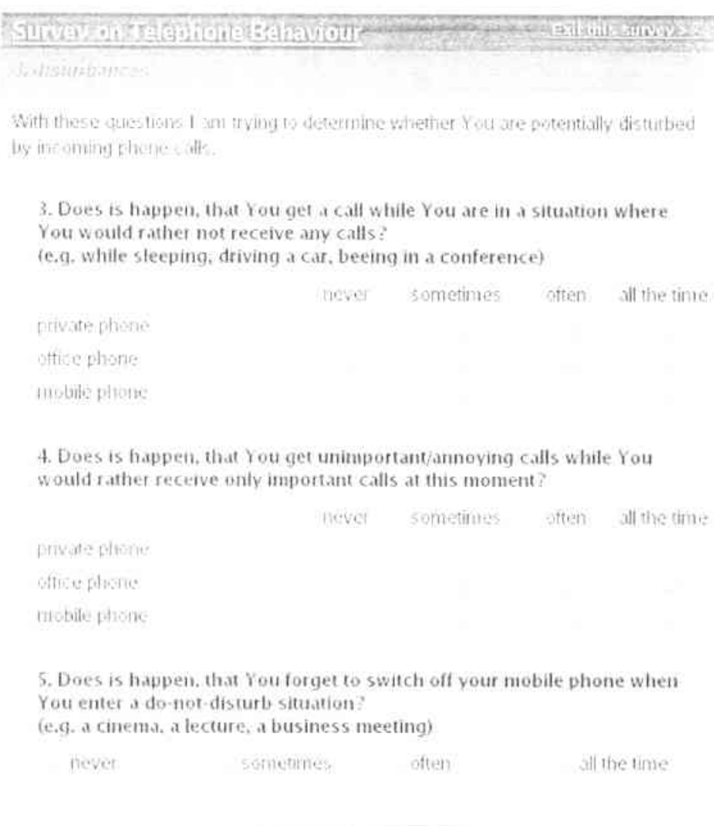

Figure 5.1: a screenshot of the web based survey

The first group of questions was concerning the respondents and their backgound.

Question one was about the current occupation of the respondents, possible answers were *working in service or consulting, working in developement or science, working in management* or being *student*.

Question two asked for the approximate number of calls received. It was differentiated between calls received on private phones, office phones and mobile phones. Possible answers ranged from *none at all* to *more than around 10 per day* respectively more than 71 calls a week.

The second group of questions was concerning the potential disturbances caused by incoming calls.

Question three asked for the frequency of calls received in inappropriate situations like *driving a car* or *attending a conference*.

Question four asked for the importance of calls received and the desire to filter unimportant calls in certain situations. Question five was "Does is happen, that you forget to switch off your mobile phone when you enter a do-not-disturb situation? (e.g. a cinema, a lecture, a business meeting)". Possible answers for all three of the questions were *never*, *sometimes*, *often* and *all the time*.

The third group of question was concerning the conceivable remedies.

Question six was about giving the caller context information about the callee. It was asked whether the respondent reckoned that people would disturbe him less often if they were aware of the current context of the callee. Possible answers were *yes, definately*, *maybe some of them would* and *most people would not care*. Additionaly the responant could mention that he would not want foreign people to know about his current context or that he would *not want to know anyone* about his current context.

Question seven asked people to imagine that there was a device that could recognize undesired calls and block them. Would they allow such a device to route undesired calls to their voice box? Answers could range from *Yes, that would be great* over *In some situations, this would be helpfull* to *No, I'd rather be disturbed than to trust a machine to decide about my calls* or *No way, any call, no matter how unimportant deserves my personal attetion.*

Question eight asked for the effort people were willing to make to adjust such a system to their personal requirements. Possible answers ranged from *I would invest almost any amount of time, if it works.* to *If it does not work instantly, I don't want it.* with options at *one minute for each address book entry*, *five minutes in total* and expecting the system to learn automatically.

Question nine explained the necessity of the system surveiling its users to determine their current context. It was asked if this would concern the respondents. The possible answers were basically yes and no with the option to limit the system to an office only environment.

The tenth and final question was wheter the respondents would be willing to pay for such a service. Possible answers were *Yes, if it works, definitely!*, *No, that's not worth paying for* and *Maybe, I'd have to try it first. But I doubt that it will work.*.

5.4 Results

The resulting answers were partly surprising in parts. The most significant result was that people seem not to forget to switch off their mobile phones in certain situations (question five). 50% of all respondents claim to never forget to switch it

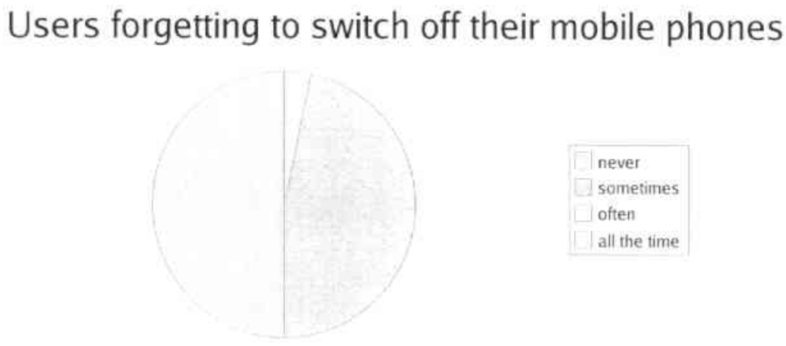

Figure 5.2: 95% of all users claim to not foget to switch of their mobile phones often or all the time

off, additional 45% do forget it sometimes. This leaves a minority of only 5% of all respondents to forget it often, no one claims to forget it all the time.

Another surprising result is the low number of total calls received. Two out of three claimed to receive only up to arround one call per day on their home phone as well as their mobile phone. The same goes for the office phones for the working people, except for those working in management and consulting. With 10 respectively 5 calls per day these two groups reported by far the highes rate of incoming calls. (These results should be treated with care as there were only 4 respondents from these groups) All in all 83% reported to get 3 calls or less per day on their office phones and even 88% claimed that for their other phones.

It also seems that student are easier to disturbe than professionals. While both groups reported the same number of disturbing calls on their private phones, 31% of the working people claim to *never* be disturbed on their private phone while none of the students gave this answer. The results for the mobile phones resembled these resulte even though the difference between students and professionals was not that significant here.

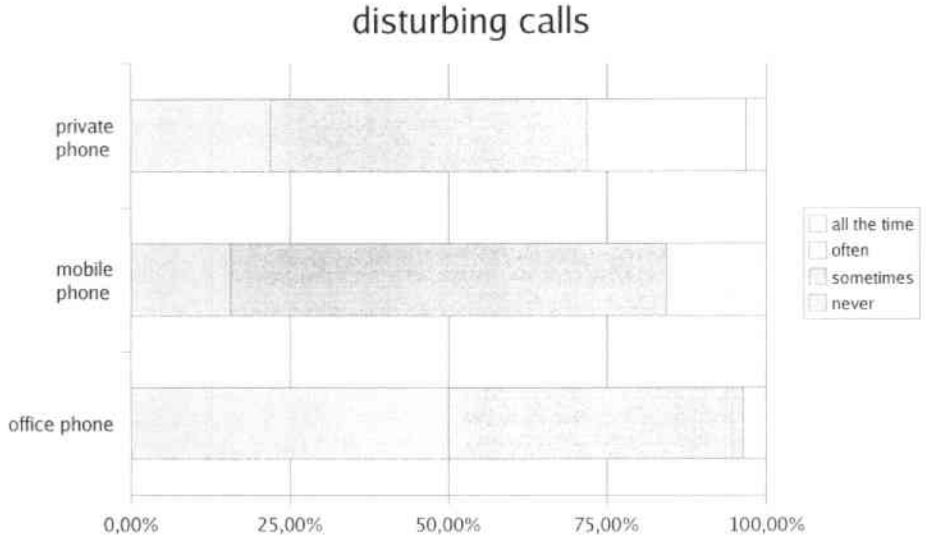

Figure 5.3: only 4% of business professionals report calls on their office phones to be disturbing often

One might argue that the home phone is the office phone of students but 37% of the business professionals also reported that they are never disturbed by calls on their office phone. All in all only one person (4%) reported to be disturbed *often* by the office phone, none reported to be disturbed *all the time*.

While these results resemble those for *unimportant calls* on the home phone pretty well and still quite well for office phones, the results for the mobile phones differ. 44% report to never get unimportant calls while they would rather receive only important calls on their mobile phones at this moment, additional 50% claim to get such calls only sometimes. This leaves only 6% of people who get such calls often, none reported to get them all the time.

Calls received on the office phone are rated as never or just sometimes disturbing in more than 90% of all cases. Contrary to this, 28% of all participants rated calles on their private phones as often or allways inappropriate, 22% are rated as unimportant at the wrong time.

Astoundingly, 16% rated calls on their mobiles to happen in inappropriate situations but only 6% rated calls as unimportant. Furthermore 44% rate calls on their mobiles as never unimportant, additional 50% reported this to happen only sometimes.

Another notable thing is the acceptance of the required surveillance. In general half of the respondents would accept the required monitoring by the system. But while the student group is willing to accept it despite their concerns in an office-only use, 25% of the professionals think that surveilance is not acceptable at all. On the other hand, 25% of the professionals had no concerns at all – contrary to the student

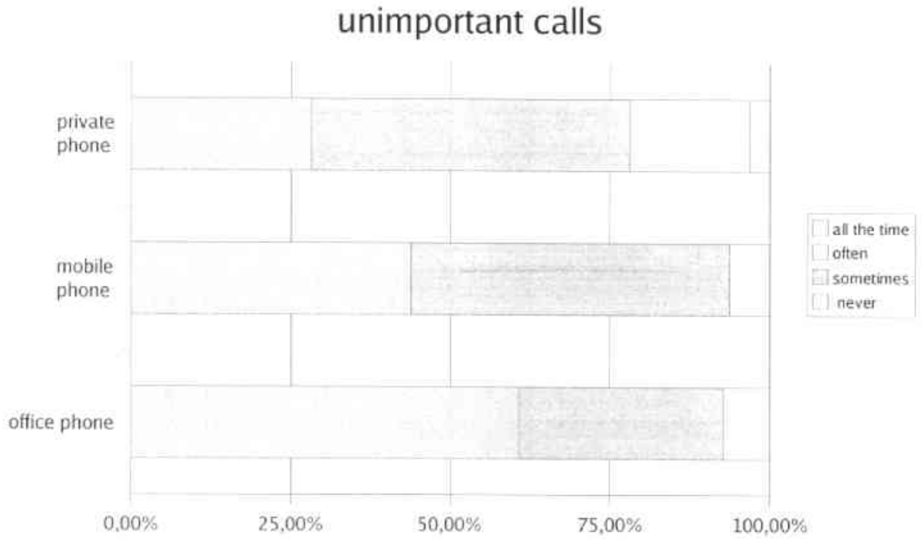

Figure 5.4: calls on mobile phones seem to be quite important, only 6% report calls to be unimportant more often than "sometimes"

group where only 12% had no concerns at all.

5.5 Discussion

First of all one has to remember once more the very limited audience on this survey. Therefore the results have to be handeled with care. This limitation goes not only for the number of participants, but also for their selection. The Connector is meant for busy professionals with a high amount of phone calls and incoming messages. Naturally those people are most busy and have the least time to participate in "unimportant" surveys of students.

The high rate of people minding to turn their mobile phones off when not wanting to be disturbed seems to show, that people got used to doing so. Therefore creating a service that automates this action on mobile phones could not be called an actual *need* but a *comfort feature*. It is arguable if an automatic system that is potentially likely to be erroneous could provide such a plus in comfort. On the other hand, half of the people forget to switch off their mobiles at least sometimes. In addition to this it can be argued if those people that remember switching their devices off are happy with what they do. A system releaving people from minding such things would be great. It would also enhance human to human communication as a mobile device that is switched off does not receive any information, not even urgent or important ones.

Another indicator for the need of mediated human to human communication is

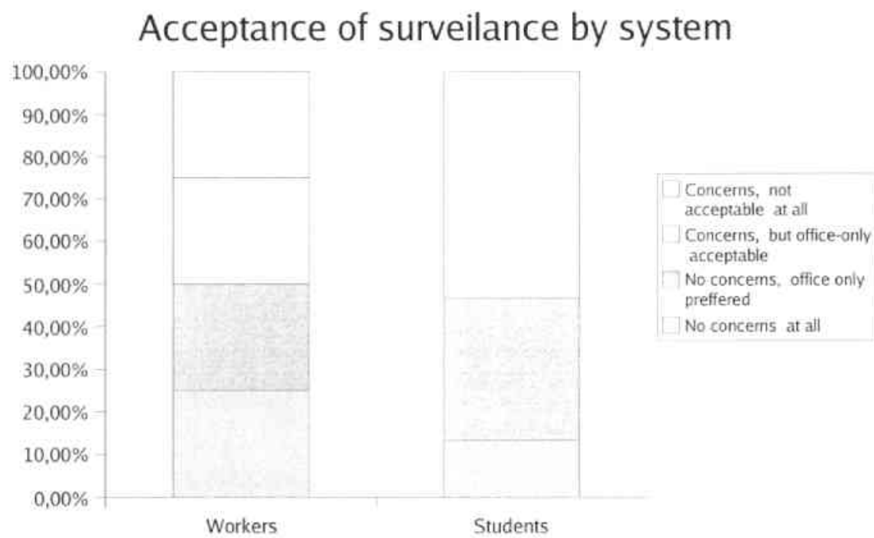

Figure 5.5: students seem to be less concerned about their privacy than office professionals

the low number of calls per day. This might be due to people's frustration about the problems with synchronous communication. As a result they evade synchronous communication and use asynchronous means of communication like mail, SMS or instant messaging instead.

As mentioned before people rated most calls received on their mobile phones as important. There are at least two possible explanations for this. The first one is quite simple. People know that calls to mobile phones are much more expensive than calls on fixed line phones. So if they try to reach someone on his standard phone but can not reach him there, they do not try to contact him on his mobile phone if the matter to be spoken about is unimportant. Another possible explanation is that people receive most calls on their mobile phones while being *on the way to somewhere*. Nowadays this normaly means sitting in a car or train or walking somewhere. Now these kinds of activities tend not to be exhaustive in their use of brain capacity. So people rate calls received while just being on their way to be not unimportant simply because they have nothing more important to do at that time. Should this be true, there are two possible conclusions to this. The first is to think about ways to use people's brains while they are on their way to somewhere by developing tasks, techniques or devices enabling them to be productive. The alternative is to explicitly use times like that for comunication and to schedule calls for times known for the user to be on the way.

A quite confusing fact is peoples impression about disturbing calls versus their acceptance of a helping system. While most people report to be disturbed by calls at home significantly more often than on their office phones, still most people would prefer a system that is restrited to an office-only environment. In other words: they want a system to get rid of disturbing calls, but only in places where they tend to not get disturbing calls. It might be a matter of marketing to get people to accept such systems in their private homes. On the other hand this also is a good "excuse" for developers to concentrate on the detection of standard office situations. These are far more likely to be detected and interpreted correctly due to the far more limited number of possible situations as well as the more standarized environment. And a good working system might be the best selling argument for people with an aversion for such systems in their home environments.

5.6 Conclusion

One of the main objectives of this survey was to determine people's need of a comunication assistance system. It is generally asumed that people are disturbed by ringing office-phones. The finding of this survey does not support this assumption. Contrariwise it seems to indicate that most disturbances by phones emerge at people's home environment, thus such a disturbance would not be one that reduces people's working efficiency but disturb their privacy. Should these findings be proven to be right, there should be a rethinking about the construction of comunication assistance systems.

It is likely that people might misjudge the impact such a system would have on their daily life and their comunication habits if asked for such things in a survey. This also goes for their judgement on how disturbing a call really is. Maybe a person that never had the chance to work in peace and silence for an extended periode of time has no idea of how efficient he *could* be, if left undisturbed. But given the very low number of daily received phone calls, this seems unlikely. It is more likely that disturbances come from other sources, like persons coming into the office themselves instead of calling. An advanced system should try to target these kinds of problems, too.

All in all one should not forget the quite limited audience of this survey, limited not only in the number of participants but also in their selection. Before beginning any work based on these findings, it will be necessary to take a much closer look on the facts related to the improvement. This survey should only help to get an idea where to look and what to ask. For future surveys one should keep in mind the target audience of the system to be designed. It will also be necessary to conduct an experiment testing the implemented protoype system in a real world situation to check its ability to ease human to human communication.

Chapter 6

Further Research

The only way to discover the limits of the possible is to go beyond them into the impossible.

Arthur C. Clarke
— *Technology and the Future*,
1972

As mentioned before, the implementation represents only the basis of what could be done. There are virtually countless improvements that could still be implemented. In the following some of them are mentioned.

6.1 Future Work

At the moment it is irrelevant for *how long* a user is in a certain situation, the only thing relevant is in *which* situation he is in currently. It is conceivable that the system's accuracy can be improved significantly by implementing *attentiveness functions*. This would consider the fact that the availability of the user changes significantly throughout the duration of the current task, e.g. if someone is reading, a disturbance within the first two minutes might be much less disturbing than after ten or more minutes. On the other hand, while having a conversation it might be much more unpleasent to be disturbed within the first minuten than after ten or more minutes. These attentiveness functions would have to be different for each situation and they would need to be pinpointed experimentally for each kind of situation and might need additional adjustment for the individual user.

Right now the shield does not take into consideration the current density of interruptions. This way it could happen, that a user is busy with a high priority task that makes his connector agent block all incomming interruptions. Taken alone, each single interruption might be not important enough to disturbe the user but taken together a high density of tried interruptions might indicate a crisis-situation that requires special attention. A user would miss this crisis if the connector agent does not inform him about unusual peaks of interrupt density. To avoid this the connector agent could create certain expectation values from the history of the user. A significant deviation from this value could trigger a special alarm.

Right now a caller is querried about the importance and urgency of his call each time he calls. There are quiete some constellations in which this is unnecessary, e.g. if the user's importance is too low to disturb the user indipendently of if he classifies his call as important or not. On the other hand the user might be involved in a very unimportant task or be idle so that any call would be routed through to him anyway. These facts could be checked before the caller is queried about his call so that this additional (and time-consuming) interaction with the system could be avoided if possible.

6.2 General/Further Ideas

While the above-mentioned improvements seem quite simple, the following ideas represent a class of improvements that are mostly not straight-forward but would require further consideration and/or research. They are presented to give the reader an impression of what could be possible and in which directions improvements are imaginable and possible.

voice recognition As mentioned in 3.6 it could be of interest which persons are currently located in the vicinity of the user. As it is possible to identify person by their voice [20], it might be possible to develope a software simmilar to the one mentioned in [14] that recognizes persons close by instead of sourroundings.

topic relation tree The topic of the user's current task is an important indicator for the relevance of a potential interruption. While at the moment it is only checked whether a call has the same topic or not, a system that shares a considerable knowledge-base about his owner could calculate a value describing the *distance* of one topic to another one. E.g. an aeroplane-engineer that is currently working on the wing-layout might be willing to be interrupted by a question concerning the fuel-storage (as fuel is stored inside the wings anyway) but on the other hand a request for some documents concerning the organisation of the next board meeting should better be delayed.

machine learning In [5] it is suggested to focus machine learning on the surrounding of the sender, as their results showed that it had by far the widest influence on the behaviour of a sender. The most interesting factors here will be the distance of sender and receiver in a *social network*.

automatic researcher The identity of a caller is important for the assessment of an incoming call. It will happen quite often that a person calls that is not in the user's address-book. It is conceivable to create an *automatic researcher*. This would allow the system to trade off the relevance and importance of a call even from a currently unknown contact, e.g. an insurance-agent could be found in a public phonebook (like the Yellow Pages), and as he calls during office-time, he most probably wants to talk about insurances.

locating agent service Sometimes it could be interesting to locate a moving user without having gps-data at hand. By knowing its origin and destination along with the start-time, an online-service might be able to go through e.g. timetables of public transportations and in this way guess the current user's location and it's proximate time to arrival.

server location The more sohpisticated the connector agent becomes, the more knowledge about the users and the owner needs to be stored. For data security- as well as for privacy reasons it will be necessary to think about the physical location of the system/server as well as about its connection or integration in certain networks. A solution that relies on large data processing service centres might not satisfy the customers need for privacy and a solution that is based on a device that is located at the users home or office might not satisfy his need for security.

Possibly a hybrid approach involving data encryption might be the right place to start looking.

request a call now In some situations, it might be expedient to have someone call you right now. This could not only be used as a getaway-excuse on a cocktail party. A much more reasonable application for a user to notify a certain group about his current availability. This will come in especially handy when the user knows that he will soon be unavailabe for some time, e.g. because he is soon to board his plane.

It seems that this service was already thought of by the 3GPP [11] and the Open Mobile Alliance [12] and is included in one of the standards they are developing. Nokia also mentiones this feature in the description of *Nokia Presence* [13].

data mining on history By browsing through the history of a user, a data mining algorithm might be able to develope new *situations* that allows the system more accurately to react appropriately.

Right now all possible situations have to be setup manually. This has two downsides. First of all it is time consuming and annoying. The second one is the fact, that a user would model situations in a way he expects his daily life to be. But there might be quite large differences in between the user's expectation and reallity, e.g. a user might oversee the fact, that he spends 10 minutes to walk to the cantina every day (and most probably also 10 minutes back).

Data mining algorithms are used to discover patterns in huge collections of data. As they are able not only to classify certain patterns, but also to find associations and sequences [21], they seem fitting for this purpose.

This way it would not only be possible to find new situation classifications but also to extract certain rules that could help to improve the systems behaviour, e.g. *85% of all international calls take longer than 20 minutes and should be re-scheduled if the lunch break is less than 30 minutes up.*

This of course requires storing all available sensor data in a user-specific history (which could become quite large) and the appropriate ammount of computing power and time to discover these possible patterns. It would also require an advanced user interface, to keep the system understandable to the user. A virtual assistent (like the helper-daemon in Microsoft's current Office-Release) could make suggestions like "It seems that you do not like to get international calls shortly before the lunch break. Should I reschedule such calls for the future?".

complex situations are the interesting ones As allready mentioned in Chapter 2 the standard situations are not the interesting ones. The detection of these situation is quite simple (at least in comparisson to the complex ones) as are the actions necessary to react in these situations, e.g. *sleeping* or *meeting* are easyly detectable and disturbances will be rescheduled if they are not urgent and important.

It starts to get interesting at the point where more than one (or two) people are involved. Not only is the detection of these *non-standard* situations challenging but the reaction to these situations is all but obvious.

For example if three colleagues are discussing a complex matter in the hallway to their offices, a machine will have great difficulties to correctly classify this situation not to speak of the right classification of the topic. If one of the participants gets a call on his mobile phone now, the system would have to not only identify his current context but also his *role* in the conversation. Is he really participating or just walking next to the other two? And if he is participating but his superior is calling him, how should the system react? This kind of questions require human intuition, at least to a certain degree.

integration of all devices It is important to include all meens of communication into the connector system to make it an efficient help for the user. Leaving one out makes obviously no sense, e.g. integrating mobile phone, mail client and instant messenger but leaving fixed line telephone out will keep the doors open for disturbances by the "unprotected" phone.

integration into existing devices It is nice to have highly sophisticated sensor components that run almost perfectly under laboratory conditions, but one must not forget, that the rank and file will just have "standard equipment". All technologies developed should be integratable into the widespread spectrum of existing devices [1]. Only this increases the chance of this system ever to be

[1] most of the current mobile devices are based on Windows CE or Symbian OS (potentially with a S60 user interface) and provide a java runtime environment

accepted by a majority.

One of the most important things to keep in mind seems to be to provide a plugin for Microsoft Outlook [2]!

[2] most of the technologies that were released lately have one and I have never heard of a new technology that found its way to mass-markets without providing at least a possibility to synchronise data with Microsoft Outlook

Chapter 7

Summary/Conclusion

If we have learned one thing from the history of invention and discovery, it is that, in the long run - and often in the short one - the most daring prophecies seem laughably conservative.

Arthur C. Clarke
— *The Exploration of Space*, 1951

In this work the idea of a personal connector agent, that manages all kinds of incoming communication and protects the user from undesired disturbances was illustrated. The goal is to create a communication device or service, that is as unobtrusive as possible. As shown, the context of the user is the key to solving this problem. The requirements for an unobstrusiv service were examined from different points of view. Furthermore, a number of possible data sources was mentioned that could help in acquiring the context of the user.

As a proof of concept and developement basis, a simple connector agent was implemented. It is able to process incoming calls, consider the current context of a user and decide to route the call to the callee, query the caller for more details and give a feedback to the caller.

Finally, several ways and ideas are mentioned to improve the connector agent. These ideas are not specifically meant for the mentioned implementation, but are rather general.

After all, my opinion is that the mentioned concepts are usefull and desirable. The importance of communication will not cease in todays society as will the desire to be undisturbed. As the means of communication are getting more and more complex, an ease for the users is required.

The industry has already spotted this problem and is also working on solutions. So far, a framework was designed that allows all kinds of reactions to a certain user's context provided that this context is known. Current solutions seem to rely on the users to adjust their current contexts themselves manually.

The automatic detection of a user's context is a highly complex matter and so far, no *perfect* solution is known, not even one, that can be called applicable and reliable for common use.

In my opinion, one should concentrate on the automatic detection of user context as well as on a uniform way to model and describe it. Constructing a whole system, including all components (detection of context, finding an according reaction and providing the necessary infrastructure) in an academic environment in competition to the industry does not make sense and is a waste of time (in my eyes) as the industry is allready settled on their approach. Of course it is still necessary to develope system prototypes to test and enhance existing ideas and develope new ones.

The context detection is the sticking point of all context aware services. There is still much to do, in a multitude of directions. One will have to keep in mind that all these solutions are ment to be used *mobile*. Even though devices a getting more and more powerfull (regarding computing power), they also are getting more and more energy consuming. Finally the infrastructural gap between highly developed areas and those with none or almost none will grow, coinciding with the further developement of the infrastructure. This will also have to be kept in mind when designing new systems.

In the end, one should not forget, that every *automatic* detection of the user's context is based on a more or less extensive surveillance of the users and their environment. This arises a whole new bunch of questions and problems, like *who* can access all this data, *what else* could and will be done with it, and in how far is the user's privacy infringed? Also there will be the question if a user *wants* to be observed 24/7. This will also be a problem of presenting the system to the public as there will always be the ones that will be reminded to Orwell's *1984* scenario – the system has to help the user, not the other way round. The successful developement of effective mechanisms to protect people's privacy will be the key factor. The human has to stay in controll of his communication.

List of Figures

4.1 the tree of the decission logic 19
4.2 The Shield GUI 21
4.3 the sensor simulator 22
4.4 the editor 23
4.5 Overall System Architekture 24

5.1 survey questions – screenshot 26
5.2 survey diagramm – mobile phone switch-off 28
5.3 survey diagramm – disturbing calls 29
5.4 survey diagramm – unimportant calls 30
5.5 survey diagramm - system surveilance 31

Bibliography

[1] Mark Weiser. The computer for the 21st century. *Scientific American*, pages 94–100, September 1991.

[2] M. Satyanarayanan. Pervasive computing: Vision and challenges. *(IEEE) Personal Communications*, 8(4):10–17, August 2001.

[3] L. B. Mummert, M. R. Ebling, and M. Satyanarayanan. Exploiting weak connectivity for mobile file access. In *SOSP '95: Proceedings of the fifteenth ACM symposium on Operating systems principles*, pages 143–155, New York, NY, USA, 1995. ACM Press.

[4] Joelle Coutaz, James L. Crowley, Simon Dobson, and David Garlan. Context is key. *Commun. ACM*, 48(3):49–53, 2005.

[5] Laura Dabbish, Robert Kraut, Susan Fussel, and Sara Kiesler. To reply or not to reply: Predicting action on an email message. Technical report, Human-Computer Interaction Institute, School of Computer Science, Carnegie Mellon University, January/February 2004.

[6] Thomas Erickson and Wendy A. Kellogg. Social translucence: an approach to designing systems that support social processes. *ACM Trans. Comput.-Hum. Interact.*, 7(1):59–83, 2000.

[7] Gary M. Olson and Judith S. Olson. Distance matters. *Human-Computer Interaction*, 15(2-3):139–178, 2000.

[8] Mike Schneider, John Graham, Aaron Bauer, Katie Bessiere, and Cleotilde Gonzalez. Shared mental models in military command and control organizations: Effect of social network distance. In *Proceedings of the Human Factors and Ergonomics Society 48th Annual Meeting*, New Orleans, September 2004.

[9] James M. Hudson, Jim Christensen, Wendy A. Kellogg, and Thomas Erickson. "i'd be overwhelmed, but it's just one more thing to do": availability and interruption in research management. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 97–104, New York, NY, USA, April 20-25 2002. ACM Press.

[10] James Fogarty, Jennifer Lai, and Jim Christensen. Presence versus availability: the design and evaluation of a context-aware communication client. *Int. J. Hum.-Comput. Stud.*, 61(3):299–317, 2004.

[11] The 3rd Generation Partnership Project (3GPP). Website, 2006. http://www.3gpp.org/.

[12] Open Mobile Alliance. Website, 2006. http://www.openmobilealliance.com/.

[13] Nokia Corp. Nokia presence technologies. Website of Nokia, April 2006. http://www.nokia.co.uk/nokia/0,,71155,00.html (english), http://nokia.de/de/64/144090.html (german).

[14] R.G. Malkin and A. Waibel. Classifying user environment for mobile applications using linear autoencoding of ambient audio. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, volume 5, pages 509 – 512, March 2005.

[15] M. Danninger, G. Flaherty, K. Bernardin, H. K. Ekenel, T. Köhler, R. Malkin, R. Stiefelhagen, and A. Waibel. The connector: facilitating context-aware communication. In *ICMI '05: Proceedings of the 7th international conference on Multimodal interfaces*, pages 69–75, New York, NY, USA, 2005. ACM Press.

[16] Nokia Corp. Nokia sensor. Website of Nokia, April 2006. http://www.nokia.co.uk/nokia/0,,73651,00.html (english), http://www.nokia.de/de/service/software/sensor/startseite/168744.html (german).

[17] Daniel Siewiorek, Asim Smailagic, Junichi Furukawa, Andreas Krause, Neema Moraveji, Kathryn Reiger, Jeremy Shaffer, and Fei Lung Wong. Sensay: A context-aware mobile phone. In *ISWC '03: Proceedings of the 7th IEEE International Symposium on Wearable Computers*, pages 248 – 249, Washington, DC, USA, 2003. IEEE Computer Society.

[18] asterisk.org. Asterisk - the open source pbx! Website of asterisk.org, April 2006. http://www.asterisk.org.

[19] Christopher Plaue, Todd Miller, and John Stasko. Is a picture worth a thousand words?: an evaluation of information awareness displays. In *GI '04: Proceedings of the 2004 conference on Graphics interface*, pages 117–126, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2004. Canadian Human-Computer Communications Society.

[20] Juha-Pekka (JP) Toivonen. Biometrics with focus on voice recognition, 2001 or later.

[21] R. Agrawal, T. Imielinski, and A. Swami. Database mining: A performance perspective. *IEEE Transactions on Knowledge and Data Engineering*, 5(6):914–925, 1993.