TNO Informatie- en Communicatietechnologie

Eemsgolaan 3
P.O. Box 15000
9700 CD Groningen
The Netherlands

www.tno.nl

T +31 50 585 70 00
F +31 50 585 77 57
info@ict.tno.nl

TNO report 33879

Finding context aware functional requirements

Date April 2006

Author(s) Leendert van Achteren

Number of pages 92
Number of appendices 15

Finding context aware functional requirements

Groningen, April 2006

By L. van Achteren

Under supervision of:

TNO Information and Communication Technology:

Ir. A. de Vries

Ir. E.J. Jager

Rijksuniversiteit Groningen:

Ir. F.B.E. van Blommestein

Dr. T.W. de Boer

Preface

Writing this thesis was unlike anything I did before. Working on a single document full-time for six months was a lot longer than working on earlier projects during my study. I liked it all the way. And that was due to a couple of things. Not only to the interesting subject. But it was also due to the pleasant colleagues at TNO ICT in Groningen. I would like to thank Arnout and Edsger for giving me the possibility to write this thesis and for giving me inspiration and support.

Writing a thesis is an intellectual task for a single individual. The thesis will be valuated on its content and its academic value. I would like to thank Fred van Blommestein for supervising and valuating my work through his own eyes, while allowing me to write my own thesis and assisting me in that process. And I would like to thank Ruben Cijsouw and Thomas de Boer for their constructive feedback.

Writing a thesis is being busy on a single subject for approximately forty hours a week, and thinking about it a lot of time besides these forty hours. A lot of time besides those forty hours, I was busy with completely other things. I would like to thank my girlfriend Lizette for supporting me in all my activities and for being the way she is.

Writing a thesis is the final step of finishing my master level education. I would like to thank my parents for supporting and directing me towards the way I am. Not only because they made it possible for me to do the things I did and do, but especially because they gave me the ambition to reach up.

Abstract

At TNO ICT in Groningen I investigated how useful context aware functional requirements can be found in a structured way. As no method to support this process already exists, I developed a method myself. In order to be able to do that I looked at what context awareness is and developed the method from there. Almost everything can be seen as context, but not everything is relevant context. Dey therefore refers to context as any information relevant to a user. Context aware applications use this relevant context information to support the user in his task.

The method

Because something is relevant to a user depending on the user's task, the method starts with a user definition, followed by a task analysis. Such a task analysis is called a scenario. In order to find functional requirements that are useful in different contexts, and in order to determine whether found functional requirements are useful in different contexts, not one but more scenarios are defined.

The advantage of context awareness can be classified in three groups. The primary functionality of such a context aware application, the (primary) functional requirement, can be found using these three classifications.

1. Context awareness can help information and services to be presented to the user according to the current context.
2. Context awareness can trigger automatic execution of a service when in a certain context.
3. Tagging of context to information can support later retrieval of this information.

The functional requirements are found by looking at the scenarios with theses advantages in mind. Afterwards, functional requirements are compared to find functional requirements that are aware of generic contexts.

Conclusion

The proposed method for finding context aware functional requirements supports a developer to find context aware functional requirements for a certain user. The method supports him by helping to define the user in a way that functional requirements can be found, and supporting him with the search for these functional requirements.

Contents

Part 1 Introduction

1 Introduction

TNO is an organization where new technologies are made available for use. “Context awareness” is such a new technology. Employees of TNO often see and hear the term in literature, on conferences and in projects that TNO has a role in. But within TNO there is little knowledge about what context awareness is and how it can be put to use. The department of “User Centered Innovations” (UCI), of the Business Unit “Mobile” of TNO ICT wants to find out what context awareness is, what the benefits of context awareness are and how they can use it to create future mobile applications.

1.1 Research formulation

This research project will contribute to the knowledge of context awareness within TNO ICT. It will do this by presenting not only information about context awareness, but also by presenting a method on how to find useful context aware functional requirements for applications. Because one of the first steps of presenting such a method will be to have a closer look at context awareness, the research objective will be the following:

To present a method for finding useful context aware functional requirements

The result of this research will answer the following research question:

How can useful context aware functional requirements be found?

Although the research takes place in an organization where a lot of mobile applications are developed, the presented method should be useful for finding context aware functional requirements for all applications.

1.1.1 User centered

As this method will be developed within the context of the department of User Centered Innovation of TNO, it is preferable that the presented method will also be user centered. User centered means that the user is seen as the most important aspect in applications. And therefore applications are built around the user.

1.1.2 Useful

With useful I mean that the user wants to use the application if he had the choice to use it or not.

1.1.3 'Context aware' and 'functional requirements'

'Context aware' and 'functional requirements' will be discussed extensively in the chapters four for 'context aware', and seven for 'functional requirements'.

1.2 Research boundaries

As this research also is the final thesis for the study "Technische Bedrijfswetenschappen" of the University of Groningen, some special boundaries occur. The research will be done with the following restrictions:

- The research will take place within the context of the department of User Centered Innovations of the Business Unit Mobile of TNO ICT
- The results must have practical relevance for TNO ICT
- The research must have an academic argumentation
- The research must follow the restrictions as stated by the Faculty of Management and Organization of the University of Groningen for writing a final thesis in "Technische Bedrijfswetenschappen"

1.3 Characterization of research

Although this research took place outside the university, this research can be characterized as a scientific research. The research is aimed at knowledge development at a specific subject.

1.3.1 Generalization

The subject of this research is the development of context aware functional requirements. It is written however from a perspective of an organization where a lot of mobile applications are being developed. Therefore extra attention is paid to the development of context aware mobile applications, but always with the objective to find a way how to develop context aware functional requirements for applications whether these applications are mobile or not.

1.4 Research methodology

In order to design a method how to develop context aware functional requirements for applications with a special interest in mobile applications, I will first have a closer look at mobile applications and context awareness. I will define them and look at the characteristics and advantages of context aware applications. Then I will present the complete methodology itself. At last I will validate the method by working out an example. This example will also figure as illustration in the earlier chapters. These three subjects will be discussed in three parts in this thesis. In order to get an answer for the main research question, I formulated a number of sub questions, each corresponding with a chapter. The sub-questions in the first part are predefined; the sub questions in the following chapters were defined based on findings in previous chapters.

The sub questions with their corresponding chapters are:

1 Introduction

(1 Introduction)

2 What is TNO?

3 What are mobile applications?

4 What is context awareness and what is its advantage?

2 The Method

5 How can context aware applications be developed with a user in mind?

6 How can scenarios be defined out of a user?

7 How can useful functional requirements be found out of scenarios?

8 How can applications be developed out of context aware functional requirements?

9 Resuming, what steps have to be taken to find context aware functional requirements?

3 Validation

10 How does a worked example look like?

11 How does the method function in the worked example?

12 What is the difference between finding context aware and regular functional requirements?

13 What is the difference between finding context aware functional requirements in a structured and an ad-hoc way?

4 Final words

14 What conclusions and recommendations can be made?

15 What is the relevance of the report?

1.4.1 Sources of information

As people at TNO ICT want to get more knowledge on context awareness and do not have it themselves, most information will be found outside the organization in papers, in books and on websites. There is however a lot of knowledge about related subjects as mobile applications and user centered innovation. The knowledge will often not be exactly what I need for this research, but due to this knowledge, it will be very useful to discuss the new found information with people at TNO ICT.

1.5 Writing of the report

In this introduction I would like to give the reasons for some choices I made in writing this report.

1.5.1 He (m/v)

In this report I often refer to people. Examples are a ‘user’, a ‘developer’, or an ‘analyst’. Where necessary I will define these people, but the point I want to make here is that I refer to them with ‘he’. Of course all these and other people I refer to in this research can be either male or female. As it is inconvenient every time to refer to somebody with noting that the person can be either male or female, I do make that statement only here and will use ‘he’ in the research from now on.

1.5.2 American English

The report is written in American English. The reason to write in English is that at first English is the standard language in ICT. If I would be writing the thesis in Dutch then a lot of English terms would be used after all or bad translation should have been used. The second reason is that an English writing has a larger group of potential readers like project partners of TNO, compared with a Dutch writing. The third reason is that in my opinion an academic student should be able to write these reports in the most important language in the world at this moment. This research is my last practice and my last test in writing such a large research in English during my Master education.

The reason to write *American* English is less well-considered. It is the default English language in my spelling checker.

2 About TNO

In this chapter I will describe TNO as a whole and TNO ICT in special, as the company where this scientific research is done.

2.1 TNO

In 1930, the Dutch Parliament passed the 'TNO Act' that regulates applied scientific research in the Netherlands. TNO was established by law in 1932. TNO stands for 'Toegepast Natuurwetenschappelijk Onderzoek'. The English version of their name is 'the Netherlands Organization for Applied Scientific Research'. The meaning of this name and the purpose of this governmental initiative is made clear in their mission [59].

> *TNO makes scientific knowledge applicable in order to strengthen the innovative capacity of business and government*

This mission is put to practice by doing consultancy, contract research, testing and giving out licenses on own inventions. TNO has an annual turnover of 555.8 million Euro in 2004, from which 194.5 million Euros comes from the government. In 2004, TNO had 4.900 employees.

The work of TNO covers a large part of the society. This is done by five core areas of competence:

- TNO Quality of Life
- TNO Defense, Security and Safety
- TNO Science and Industry
- TNO Built Environment and Geosciences
- TNO Information and Communication Technology

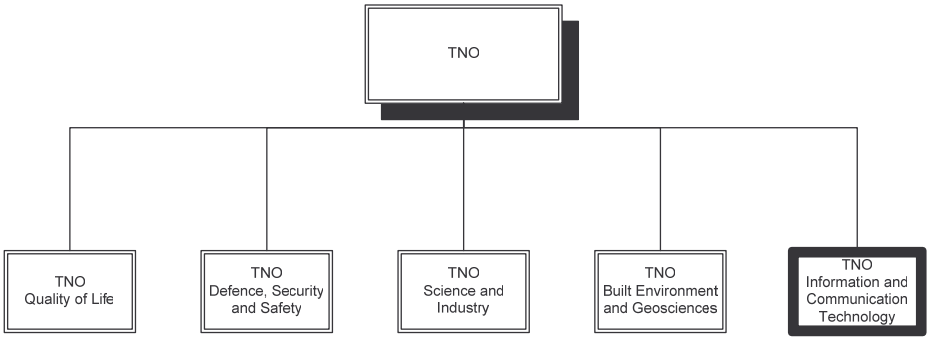


Figure 2-1 – Organization of TNO

2.2 TNO Information and Communication Technology

During the last decade of the second millennium TNO wanted to widen their view towards Information and communication technology. Some areas of competence had activities in ICT, but ICT became important enough to have a whole department working on it. In 2003 KPN, the largest telecom operator in the Netherlands decided to sell its research department. TNO grabbed the opportunity, and joined the newly acquired company with a part of the department “Physics and Electronics Laboratory”, that still existed at that time. TNO ICT was born.

2.2.1 Organization TNO ICT

As the scope of TNO ICT still is very broad, the competence area is split in four markets oriented Business Units [75]:

- BU Corporate

  The Business Unit ‘Corporate’ focuses on all non-public companies outside the telecom sector and is formed from the departments ‘Business innovation & Modeling’ and ‘Future enterprise strategies’.

- BU Mobile

  The Business Unit ‘Mobile’ focuses on the mobile market of ICT. The departments ‘Mobile Information Technology’, ‘Mobile Network’ and ‘User Centered Innovation’ are part of this BU.

- BU Public

  The Business Unit ‘Public’ has the public sector as market. The departments ‘Telecom Infrastructure and Services’, ‘Network & Information Security’ and ‘E-Business & E-Government’ are part of it.

- BU Wireline

The Business Unit 'Wireline' focuses on the wired market of ICT. The departments 'Broadband and Voice Solutions', 'Business and Network Optimization', 'Last Mile and Telephony Solutions' and 'Service Architectures and Security' are part of this BU.

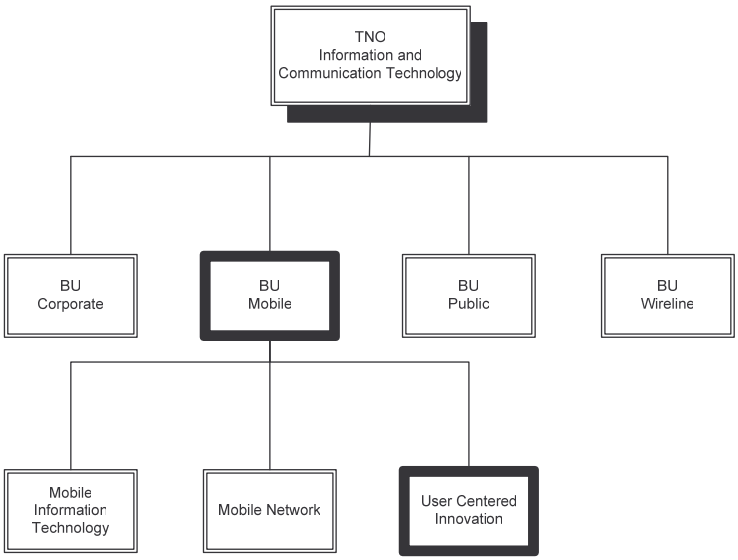


Figure 2-2 Organization TNO Information and Communication Technology

2.3 The department of User Centered Innovation

User Centered Innovation (UCI) has a team of innovators with knowledge of market, user needs, user interfaces, technology and service development work together in multi disciplinary teams. By developing the market of mobile services like i-mode, SMS, GPRS/internet everywhere, more traffic and therefore more revenues will be generated on the networks of mobile operators, which are important clients of TNO ICT. For a number of services, the added value will be given by building and adding network service-elements like location information, personalization or billing. The power of UCI lies in determining the communication needs of users and developing service concepts, building prototypes and executing pilots with users, in cooperation with partners and contractors. As the whole department is user centered, it will be preferable that the method of developing context aware applications will also be user centered.

2.4 The relevancy of this report for TNO

There are four main reasons why this research is relevant for TNO:

1 As TNO wants to make scientific knowledge applicable, it is necessary to master this scientific knowledge. Context awareness is such a technology and this research will help building knowledge about this subject in the organization.
2 TNO ICT stated seven areas where more knowledge should be gathered in 2005. Context awareness is one of them.
3 At a more detailed level, the department of UCI develops mobile services and applications. Context awareness might be a technology that contributes to making more technologically advanced services and applications. This research delivers a method to develop these context aware mobile applications.
4 Employees of TNO ICT have a role in a number of research projects. In at least two of them context awareness is a major issue:
    - Freeband: PNP2008 [76]
      Development of a user centric ambient communication environment
    - Freeband: Frux [77]
      Better cooperation and care through smart context aware group services

3 Mobile applications

The method for finding context aware functional requirements is developed for applications in general, but will often be used for mobile applications by TNO and therefore will also be validated using mobile applications. I will define a number of terms regarding these mobile applications and applications in general in this chapter. A piece about the history and current state of mobile devices is added in appendix B.

3.1 Definitions

Here I will define mobile applications. I define them as follows:

Mobile applications are applications that run on mobile devices

3.1.1 Mobile devices

The use of mobile devices is growing. In 2004 the market grew with 31% to 634 million devices [1]. This number is estimated by Nokia and based on their following definition of a mobile device [2].

A handheld that is connectable to a mobile telephone network.

I will also use this definition of a mobile device because mobile phone related companies are the core market of the Business Unit ‘Mobile’ where I operate in the department of UCI.

3.1.2 Applications

One word can have a lot of definitions, each meaning largely the same [80][81][82][83]. I used them to create my own. I define an application as:

A computer program aimed to support the user in his task.

3.1.2.1 Applications versus services

Where an application is aimed to support a user, a service could unlike an application, also support an application or another service.

a service is a computer program which does not per se directly support a user.

4 Context awareness

One of the goals of this research is to find out what context awareness is. In this chapter I will first define context and then I will add the 'awareness' to the context.

4.1 Context

'Context' has a lot of definitions. As an illustration, I will present the definition of the online dictionary Dictionary.com [34]:

- *The part of a text or statement that surrounds a particular word or passage and determines its meaning*
- *The circumstances in which an event occurs; a setting.*

Dictionary.com and others [35] define context as the circumstances or setting of text, an event, human behavior or something else. Because the complete situation of the universe at the time being and in advance can be seen as circumstances in which an event occurs, the complete context of something is unlimited. I will use "context" in a computing context, which I will define below.

4.1.1 Context in a computing context

There is a friction between the definitions where context is unlimited big, and the fact that the information processing capacity of computers is limited. Schilit and Theimer [39] in 1994 were one of the first to define context in computing. They saw location of use, the collection of nearby people and objects, as well as the changes to those objects over time as context. Pascoe defines context to be the subset of physical and conceptual states of interest to a particular entity [40]. But these definitions are too specific according to Dey [41].

> *"Context is all about the whole situation relevant to an application and its set of users. We cannot enumerate which aspects of all situations are important, as this will change from situation to situation. For this reason, we could not use these definitions provided."*

And I agree with him. The capacity of a computer should not influence the definition of context. The fact that not all context information can be used by an application is not important for the definition of context. It is the relevance of context, not the definition

of context that should determine whether context information is to be used or not. And it is the capacity to get the context information in the system to determine whether the relevant information can be used. Dey presented the following definition of context:

> *"Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves (…) where relevancy depends on the user's task."*

One important part of Dey's definition is that he refers to an application. By doing that, he implies that there must be an application being used by the user. Therefore only tasks where an application is involved can be considered. As applications can only be used on computing devices, this implies that his definition of context is one in a computing context.

4.1.1.1 User and applications

Different organizations divide the total of context information in different categorizations of context. In appendix C I compared categorizations used at Xerox Corporation and by the people at Freeband. Both organizations define the same groups that Dey also presented in his definition of context, namely application and user.

4.1.2 Context relevance

The fact that I was born in a small village called Oude-Pekela may have influenced the content of this thesis, but it is probably not as relevant as the fact that I write this thesis in the year 2005. But when you read this thesis because you are investigating where thesis writers were born, the relevance of this context information will be totally different. Relevance is very hard to predict. The relevancy of context information depends on the user's task according to Dey. So we first have to identify the user's task in order to determine the relevant context information. If a user type is presented in a way that his typology determines his tasks, this confirms that the decision to start the development of a context aware application with a user type is a right one. In the next chapter I will have a closer look at the relation between the relevancy of context information and a user task.

4.1.3 Context availability

Another problem of context is that the context information is not always available. A person can only do something with context information that is available [36]. The University of Helsinky defines context information as available information about

context and also emphasizes that almost any information can be seen as context information:

> *Almost any information available at the time of interaction can be seen as context information.*

If context information is necessary for an application, it is important to make sure that the needed context information is available or can easily be made available. In appendix E, I wrote how context information can be obtained by an application.

4.2 Context aware applications

Raatikainen defines context awareness as that [43]:

> *One is able to use context information".*

That 'one' is here the application. Dey defines context awareness as follows [48]:

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

The obvious difference is that Dey wants a system to use the context information to provide relevant information and/or services, where Raatikainen already is satisfied if a system just uses context information. In this thesis I want to support users by letting them use context aware applications and therefore an application must do something relevant with the context information. Therefore I will use the definition of Dey. In appendix A, as an illustration, some currently available context aware (mobile) applications are described.

4.2.1 Need fulfillment by context aware application technology

Mari Korkea-aho [58] presents three features that are characteristic for context-aware applications. He bases these features on works from Dey and Abowd [47], Pascoe, Ryans and Morse [48] and Schilit, Adams and Want [49].

4.2.1.1 Information and services can be presented to the user according to the current context.

This includes the selection of proximate information and services, and contextual commands. An example of the previous would be information on where the closest

bank is. An example of the latter would be a user interface changing commands depending on the time of the day or location.

> *Think of a scenario where you want to contact a colleague, but you don't know what the best way to contact him is. You do need to talk to him but it's not important enough to disturb him in a meeting. With a number of Nokia, Motorola and Sony Ericsson phones [50] in combination with a connection to a mobile data network [51], it is possible to see presence information about your contacts. The colleague can for example describe his preferred way to be contacted at the moment. Or he can make some presets like 'occupied', 'blocked', 'in car', 'in meeting'.*

Figure 4-1 – Nokia - my presence

4.2.1.2 Automatic execution of a service when in a certain context

This includes context-triggered actions and contextual adaptation. An example of the previous would be that when a user enters a specific room her mails would be shown on a nearby terminal. An example of the latter would be the change on volume on a phone according to the current noise level.

> *The SenSay is a context aware mobile phone that adapts to dynamically changing environmental and physiological states [52]. Its sensors for sound, light and movement give information which together with calendar information is used to define the user's current status. When a user is engaged according to his calendar, or when he doesn't have a meeting, but is speaking, his status will be 'busy'. Incoming calls will be handled in a way that suites the current status of the user.*

Figure 4-2 – SenSay – a context aware phone

4.2.1.3 Tagging of context to information for later retrieval

This includes that context information is stored together with documents, meetings, and so on.

> *Forget-me-not is a handheld device developed in 1994, which stores information together with the context of this information. If two users exchange a document for example, the device remembers not only which document is exchanged, but also with whom, at which time, where the users where at the moment, who else was there, etcetera. This way a user can later for example search for the name of the document that he exchanged with Bill at his office where also Peter was present, two weeks ago.*

Figure 4-3 – Forget me not – Context as retrieval key

4.2.2 Adaptive mobile applications

The original title of this report was “adaptive mobile applications”. The original purpose was to have a look at context awareness in mobile applications. But what are adaptive mobile applications? Is it the same as a context aware mobile application?

The word ‘adaptive’ comes from the English verb “to adapt” which means: “To make suitable to or fit for a specific use or situation“[4]. An adaptive mobile application is a mobile application that makes itself suitable for a specific user or situation. Sadjadi writes [44]:

> *“An application is called adaptive, if it’s readily capable of adapting its behavior in response to some expected or unexpected changes in its context (not requirements) to fulfill the application’s functionality as much as possible.”*

As noted above, Mari Korkea-aho presents three features that are characteristic for context-aware applications:

1. information and services can be presented to the user according to the current context
2. automatic execution of a service when in a certain context
3. tagging of context to information for later retrieval

According to Sadjadi's and Mari Korkea-aho's definitions, the first two features account for adaptive applications. It's not necessary to exclude the third feature out of the scope of this thesis. Therefore from now on I will discuss context aware functional requirements for the development of context aware mobile applications.

Part 2 The method

5 From user to context aware application

One of the goals of this report is to develop a method to find context aware functional requirements for applications with a specific user in mind. After defining context and context awareness in chapter 4, in this chapter I will describe how context aware applications can be developed with a user in mind.

5.1 Position of the method in an application development method

The method I present in this thesis describes how to find context aware functional requirements. At the end of this method no context aware applications are developed yet. In this paragraph I will describe what part of application development is covered by this method. I will use the System Development Methodology (SDM) [88], because this methodology covers the entire process of application development, from defining the targets of the new to be developed application until evaluation of the completed product. In SDM, application development is regarded as a project with a beginning and an end. But it is also seen as an ongoing process; completed applications are evaluated and these evaluations are the basis for an improved application. Seeing application development as a project with steps between begin and end is useful for positioning the method for finding context aware functional requirements. This because it illustrates very well which parts of application development is supported with the method.

In SDM seven stages are defined:

1. Information planning (IP)

   In the information planning stage the targets and the borders of the project are set in a project plan.
2. Definition study (DS)

   The definition study is used to test the project plan on technical, organizational, economical and social feasibility.
3. Basic Design (BD)

   In the basic design logical subsystems are defined.
4. Detail Design (DD)

   The detail design holds the data-, functional and technical specifications of subsystems.
5. Realization (R)

The realization is about realizing the earlier defined specifications. Not only creating a full system specification, but also building the systems and describing user tests.

6. Introduction (I)

   The introduction is about preparing the introduction and actually introducing the product.

7. Usage and management (U&M)

   In the final stage support and quality management are described and performed.

The method described in this thesis supports finding functional requirements for context aware applications. The functional requirements found are not sufficient for building the application. Further, more detailed functional requirements still have to be defined. The functional requirements found by the method define the primary context aware functionality of the new to be developed application. Realizing the most promising of these requirements will be the target of further application development. Determining which of these functional requirements are the most promising still has to be done. Also the other aspects of planning an application development project have to be done. The method presented in this thesis therefore only covers the information planning stage and then yet only a part of this stage. This is illustrated in figure 5.1.

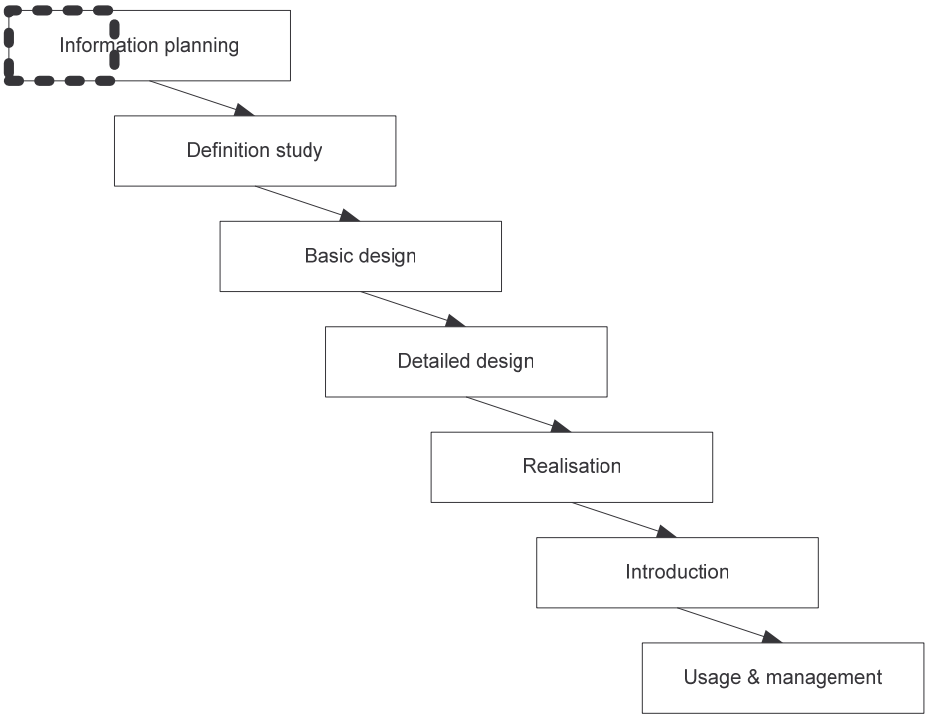


Figure 5-1 - Finding context aware functional requirements in SDM

5.2 Three steps from user to working applications

In the last paragraph I wrote that scenarios should be used to develop context aware applications. In this paragraph I will describe which two steps are supported by the method and where that brings us in the process of designing context aware applications.

Step 1: From user to scenarios
Step 2: From scenarios to functional requirements
(Step 3: From functional requirements to working applications)

In chapter 6, step 1 is described. I will first describe how a user should be defined and has to be described so that tasks become clear. I will than describe methods of choosing and working out the different scenarios and I will finally give criteria for a good task analysis method, compare several task analysis methods, choose the best one and describe it in detail.

In chapter 7, step 2 is described. I will argue why relevant context information can be extracted from the task by finding useful functional requirements and present methods to find these.

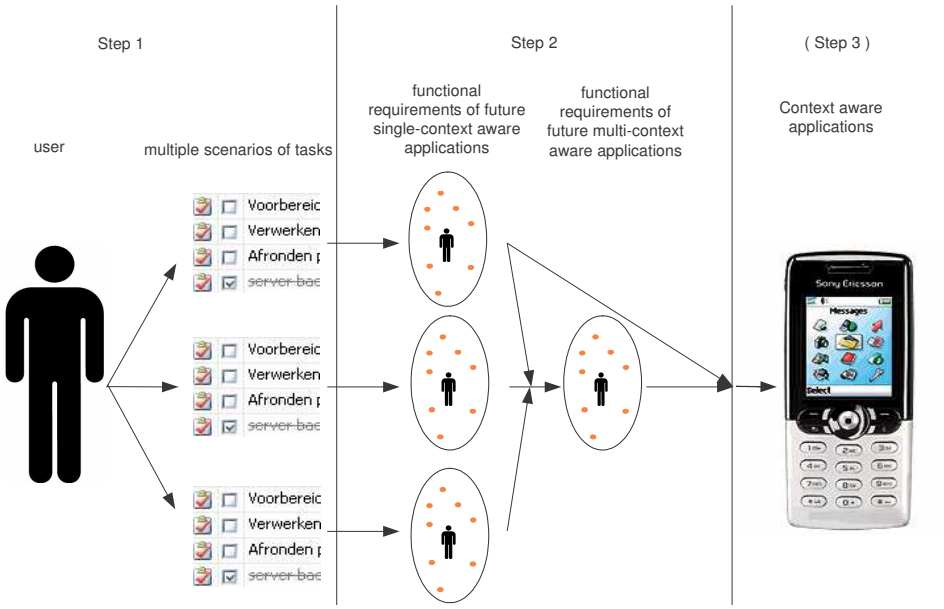


Figure 5-2 – From user to context aware applications

In chapter 8 I will describe step 3. Step 3 describes the steps to be taken to develop context aware applications after the context aware functional requirements are found using the method.

In chapter 9 I will make a summary of the steps to be taken to develop context aware mobile applications.

6 From user to scenarios

In this chapter I will start to argue why more than one of such a task analysis is necessary. Further on I will have a closer look at users and tasks. In paragraph 4 I will discuss different task analysis techniques and choose the best one for this method. In the last paragraph I will discuss the chosen technique in further detail and I will describe how the technique can be used in the method.

6.1 Why scenarios are used in the method

In this paragraph about scenarios I will discuss two reasons why it is better to use multiple task analysis for finding context aware functional requirements than using only one task analysis.

6.1.1 Scenarios should be used to find applications that are more generic and aware of more context

In a task analysis, the task of a user is described. In chapter 4 I wrote that the relevancy of context information is related to the user's task. If however a single task analysis is build from a user, the context aware applications that can be found using that task analysis might only be aware of the context relevant in that single task analysis. It would be nice to be able to say something about different tasks where the same applications may be used, but in a different context resulting in different behavior, optimized for that other situation.

The techniques offered in chapter 7 for finding context aware applications out of a single task analysis will often result in functional requirements which are aware of one or more type(-s) of context information instead of functional requirements which are only aware of the context information explicitly described in that specific task analysis. Using different scenarios and combining the found functional requirements from each task analysis can however result in applications which are more generic and are aware of a broader range of potentially relevant context information.

6.1.2 Scenarios should be used to find applications that not only are useful in a very narrow defined situation

Another problem of finding context aware functional requirements out of a single task analysis is that the resulting applications will be extremely useful to a very narrow defined user in a very narrow context. This is often not a desired situation. Because the

multiple scenarios will together describe more situations, context aware application can be developed from the resulting functional requirements that are useful in more than one situation.

6.2 User

The method starts with a user. In this paragraph I will state how this user should be defined.

6.2.1 User definition by scenarios

As became clear in the previous chapter, it is important to find out what the tasks of a specific user are in order to be able to define relevant context information based on these tasks. A user must be defined in a way that tasks can be derived from this definition. This can be done by defining the user by his task. A definition could for instance be “someone who changes a flat tire”. But as we want to build different scenarios, the user might have multiple tasks. In that case, the user should be defined by these multiple tasks.

6.2.1.1 Choosing different tasks for the scenarios

Choosing these tasks can be done through different considerations. I will now give the two main considerations to be made.

6.2.1.1.1 Using broad versus narrow task definitions in scenarios

The different tasks can be chosen to get a very broad definition of the user or they can be chosen to get a very narrow definition of the user. A broad user definition will result in a broad potential market when the product gets on the market. A narrow user definition will result in a smaller market, as the resulting applications will be optimized for a specific group of users. On the other hand, a smaller user definition results in more scenarios with comparable user tasks, which will result in better applications in comparison with a broader group of tasks. Therefore the first consideration is whether applications for a broad market are developed or better applications for a smaller market.

6.2.1.1.2 Tasks can differ from each other in multiple ways

I will get back to tasks in paragraph 6.2.4, but I will give a small preview here as the way that tasks differ is important for user definition in scenarios. Different tasks can be described by describing different activities, but different tasks can also be described by performing the same activities in other circumstances. Task 1 can for example be to call

someone and task 2 to email someone, but task 1 could also be to call someone from a car and task 2 to call someone from an office. The second consideration is therefore in what ways the tasks are going to differ. The scenarios differ from each other by activity (primary task) or by circumstances, where the goal of the task is one of these circumstances. If the goal of a task is changed, the task analysis will often not change a lot on the downside of the primary task as more or like the same subtasks will have to be performed to perform the primary task.

6.2.1.2 How far tasks should be defined in a user definition

I stated earlier that a user should be defined by his tasks. I will discuss tasks in paragraph 6.3, but regarding to user definition I would like to mention that not the entire set of tasks can be defined at this moment. The task analysis that is going to be build, forces the application developer to further define the user's task in a scenario. It is sufficient in this stage to define the user's task only by the primary task to be considered.

6.2.2 The target group a user belongs to

Whether the user is in a private or work environment, whether he is older then 65 or a teenager, these and any other target-group decisions are up to the developer to be made. These specifications define the circumstances where a task is being worked out. These circumstances are important for building an application, because:

- The properties of the group determine the frame where the tasks of the user fit in.
- The maximum cognitive load and knowledge of design standards of different groups differ.

For relevant context information however, it is not important to determine the target group, as this relevancy is only related to the task of the user. So it is important to state the target group, though not directly for determining relevant context information, but as a help for determining the user's tasks and as help for determining the maximum cognitive load of, and probable amount of knowledge of design standards by users.

6.2.3 The circumstances a user is in

As stated in paragraph 6.2.1.1.2, the circumstances that tasks are fulfilled in can be a part of the task. This way, the same activity can take place in other circumstances, resulting in another scenario. Sometimes, during the process of working out the task analyses, certain assumptions have to be made, for example about what kind of devices

the user can use. An assumption can be made for one scenario, or for all the scenarios. If the assumption is made for one scenario, it should be taken into the task analysis, or written down as an assumption for this task analysis. If the assumption is made for all the scenarios, the user is further defined by this assumption. The user definition has to be extended with these assumptions. Defining the user and building a task analysis therefore has an iterative element.

6.2.3.1 Applications and devices used in a scenario.

The applications and devices used by the user in the scenarios are part of the circumstances described above. The information about present applications and devices is valuable for finding multiple-context aware functional requirements. More about this can be found in paragraph 7.4. In the current paragraph I will describe how these applications and devices should be decomposed in functionalities [86].

First the devices used in a scenario have to be recognized. In scenario B as worked out in chapter 10, a mobile phone and a PDA are used. The next step is that the applications on these devices should be recognized by their functionality. Their functionalities are described here.

> *The mobile phone has calling functionality. The PDA can be used as an agenda. These functionalities are a part of the device, but these functionalities can in their turn have concrete functionalities in them. The agenda has a list with appointment description, begin time, duration, location and contact. The calling is done with help from a phonebook application, and a connection part. The phonebook application has an entry for "Bert Mobile".*

These findings can be drawn into diagrams [87], appendix K. Each of the blocks gets a code so that they can be referred to. The code exists of the character for the device, and a functionality number starting with 1 at each hierarchy level. An example can be found in appendix K, the functional decomposition of available devices in the worked example.

6.2.3.2 The quantity of scenarios

How many scenarios are to be worked out depends on what the developer wants to achieve. Adding a scenario results in an extra task to explore which will result in more time to invest, but also a greater spectrum of the user covered. Adding a scenario can for instance be used to investigate a user scenario that besides the already chosen scenarios also seams to be interesting. Except for extra time investment in working out

the scenario for finding functional requirements and extra time investment in the step following the steps presented in this method where functional requirements have to be chosen for further development, there are no disadvantages for adding extra scenarios.

6.2.4 User definition all together

Concluding, a user definition has to include the following elements:

1 A user is primarily to be defined by his tasks
2 Target group specifications are to be made
3 Assumptions regarding the user have to be set down

6.3 Tasks

Earlier in this research I made the relation between users and tasks. In this paragraph I will define tasks and introduce important aspects of tasks.

6.3.1 Goals versus tasks

Alan Cooper, also known as the 'father of Visual Basic', wrote some books on application design. In his works he pays a lot of attention to usability. He argues that users are the most important part in man-machine communication and that an application should be build around the user [63]. In order to build an application around a user, Cooper distinguishes goals from tasks [60]. Cooper argues that an application designer should look further than the user task. A designer should determine the goals of a user and why these are the goals of the user. This implies that when making a task analysis as input for application development, it is necessary to pay attention to the goals belonging to the task. Norman does also distinguish tasks and goals as explicitly as Cooper does [66] and he also uses task-structures in order to design an application [61]. A task as mentioned by Dey in his definition of context awareness [41] could be read as the system of current tasks of a user, including his goals. But one could also regard the goals of a user as potential relevant information about an object relevant to the user's primary task. In both cases Dey's definition implies that goals can be used to find relevant context information.

I share the distinction between goals and tasks. I will separately define my usage of both goals and tasks. When I write about a task, I mean the activity in order to reach a goal. When I write about a goal, I mean the reason why one or more tasks are being performed.

In the worked example in chapter 1 I describe for example the task that a user calls somebody. I named the user Anton and 'somebody' Bert, to make it more visual. The primary task of the user is that Anton calls Bert. The goal of this task is that Anton wants to cancel a meeting with Bert.

Later in this chapter I will have a look at different task analysis techniques and choose one that seams to fit best. One of the demands for this task analysis will be that it supports the availability of goals.

6.3.2 The hierarchy of tasks

Cooper and Norman both distinguish tasks from goals, because both are useful in developing applications. But Cooper also asks himself why a user has a certain goal:

> *"Although it is the user's job to focus on her tasks, the designer's job is to look beyond the task to identify who the most important users are, and then to determine what their goals might be and why."*

Questioning why a user has a certain goal means that Cooper is looking for the higher goal of the goal of the user's task. Apparently there is a higher goal then the goal related to the task. This implies that a goal of a task can be a task in itself, with its own goal, which in its turn can be a task again. This continues until the highest goal in the hierarchy is reached. The highest goal is to fulfill a 'basic need'. Maslow wrote about these basic human needs [19]. In appendix **Error! Reference source not found.** I set apart his theory on basic human needs.

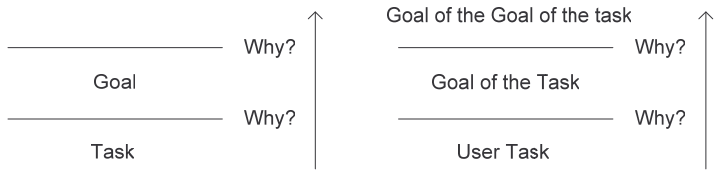


Figure 6.1 - Hierarchy of goals theory

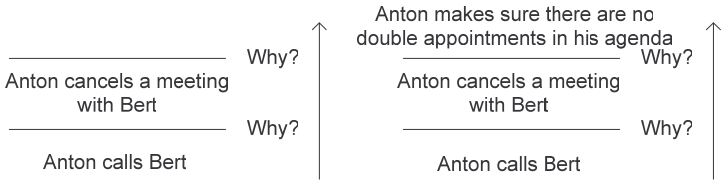


Figure 6.2 - Hierarchy of goals practice

One of the criteria for finding a good task analysis method will be that it supports the hierarchy of tasks and working this hierarchy up- as well as downwards.

6.3.3 Tasks versus applications

In chapter 3 I defined an application as:

A computer program aimed to support the user in his task.

In this research I want to find context aware functional requirements. I do this by analyzing a user's task. In this task analysis, the user can already be supported by one or more applications.

6.4 How to get from user to tasks

Task-structures are a representation of a user's system of tasks and goals. Task-structures can be built by performing task analysis. There are multiple methods to analyze tasks. I will first determine the criteria a task analysis method should meet before I will describe different task analysis techniques and at least, choose the one that best suits my needs.

6.4.1 Criteria for a good task analysis method

Cheng and Johnson did research on applying task analysis to facilitate the design for context aware applications [65]. Unfortunately they did not really do this. All they did was defining some criteria for choosing a task analysis to facilitate the design of context aware applications and then using this analysis method to describe the differences between context aware applications and non-context aware applications. They found it important that an application is easy to use by developers who have little knowledge about task analysis, both for the making of a task analysis as for getting information from it. They also stated that it should be possible to concentrate on a part of the task

system without losing the picture of the overall task. I translate these demands in the following criteria:

- A task analysis should be easy to build and read without a lot of task analysis knowledge.
- Focus on a specific part should be possible while the big picture remains clear.

I agree with these criteria, but I want to add what I found in paragraphs 6.2.1 and 6.2.2, that the task analysis method must support goals as well as a hierarchy of goals and working this hierarchy upwards as well as downwards. Finally I would like to add the criterion that a method preferably does not take a lot of time to work out, as time is scarce.

- A distinction of tasks and goals has to be supported.
- A hierarchy of goals has to be supported.
- The hierarchy of goals must be taken up- as well as downwards.
- The method must not take too much time.

As I have some additional requirements above those of Cheng and Johnson, I will choose a suitable task analysis method myself after setting apart the difference between wishes and demands.

6.4.1.1 Wishes and demands

Some criteria are absolutely necessary in the task analysis. Other criteria are very welcome. Pahl and Beitz propose to distinguish ‘wishes’ from ‘demands’ [79]. Demands are requirements that can not be left away. Wishes are very welcome, but not strictly necessary. In comparing the different task analysis techniques, I will first have a look at what methods meet the demands and then use the wishes if necessary to choose the best option.

I see the criteria about distinction of tasks and goals, support for a hierarchy of goals and the ability to work through this hierarchy top-down as well as bottom-up as demands, because they are essential to find relevant context information in a later stadium. The criteria about the process of task analyzing like how usable a task analysis method should be are wishes to me. The easier a method is to use, the more overview remains while focusing on a specific part and the less time it takes to use a specific method, the better.

Demands:

a. A distinction of tasks and goals has to be supported.
b. A hierarchy of goals has to be supported.
c. The hierarchy of goals must be taken up- as well as downwards.

Wishes:

d. Focus on a specific part must be possible while the big picture remains clear.
e. Task analysis must be easy to build and read without a lot of task analysis knowledge
f. The method must not take too much time.

Of the wishes, I see the first: “focus on a specific part must be possible while the big picture remains clear” as the most important wish, as it has the least process characteristics.

6.4.2 Different task analysis techniques

Cheng and Johnson chose a task analysis technique, but mentioned only two techniques they considered. These were the Hierarchical Task Analysis (HTA) method and GOMS (Goals, Operators, Methods, and Selection Rules). HTA suited their needs best but I will reconsider them both as I have additional criteria.

Kirwan and Ainsworth [64] compared the use of a number of task description methods. They did not do this comparison to find a task analysis to facilitate the design of context aware applications. They just described a number of task analysis methods where it is up to the reader to choose the right one for his purposes, as I will do. Kirwan and Ainsworth compared the first six following task analysis techniques. The Husat Research Institute presents a potential interesting adaptation on the Hierarchical Task Analysis mentioned earlier.

1. Charting and network techniques
2. Decomposition Methods
3. Link Analysis
4. Timeline Analysis
5. Operational Sequence diagrams
6. Hierarchical task analysis
7. Husat HTA
8. GOMS

A description of all these techniques can be found in appendix A.

6.4.3 The best technique for the job

Here I will combine the task analysis methods described above with the criteria as stated earlier. I will first see which methods meet the demands before comparing the ones that do with the wishes.

6.4.3.1 Meeting demands

Comparing the methods mentioned with the demands as stated filters four methods out. As goals are of crucial importance for my task analysis, link analysis, decomposition methods, timeline analysis and operational sequence diagrams are not suited for the job. Also regarding the demand that a hierarchy of goals must be possible to be taken bottom-up as well as top-down, only the Husat HTA adaptation remains a possibility. The "charting and networking" technique get a neutral valuation for all demands because the charts and networks are able to represent task and goals, in a hierarchy and top-down and bottom up, but the technique does not hold this criteria standard.

	1	2	3	4	5	6	7	8
Demands:								
A distinction of tasks and goals has to be supported.	o	-	-	-	-	+	+	+
A hierarchy of goals has to be supported.	o	-	-	-	-	+	+	+
The hierarchy of goals must be taken up- as well as downwards.	o	-	-	-	-	-	+	-
Wishes:								
Focus on a specific part must be possible while the big picture remains clear.							+	o
Task analysis must be easy to build and read without a lot of task analysis knowledge							+	o
The method must not take too much time.							+	o

6.4.3.2 Regarding the wishes

Because this technique is an adaptation of the regular HTA technique, I wanted to make sure that an adaptation of another technique might not be better. The GOMS technique received a positive valuation for two out of three demands, so an adaptation might upgrade this technique to meeting all demands. Therefore I compared GOMS with the Husat HTA adaptation on wishes. There I found out that the Husat HTA technique scored better than GOMS on all the formulated wishes. I therefore have no reason

anymore to doubt whether there might be a better alternative than the Husat HTA technique.

6.5 Hierarchical task analysis (HTA) + middle out

In this paragraph I will have a more detailed look at HTA in general and at the middle-out adaptation.

6.5.1 HTA

Hierarchical task analysis was introduced by Annett and Duncan in 1967 [67]. They planned to introduce some general principles for task analysis, but instead HTA is currently used as a standalone task analysis method. Shepherd wrote a book about Hierarchical Task Analysis [62]. He describes the process of HTA as a system of tasks being explored in progressively greater detail until it is understood sufficiently. “Sufficiently” is a kind of arbitrary word. It means that it is up to the analyst to determine whether the level of detail is high enough. I will come back on this subject later.

Shepherd uses HTA in a top-down way. He presents a cycle of decisions to be made in order to work out the task structure [Figure 6-1]. He starts describing a goal and then asks himself whether the goal is carried out to a sufficient standard. If not, he asks himself whether the goal can be examined usefully. This extra examination can lead to the formation of sub-goals or tasks. If not, then the goal may be described another time. If not, then the operation should be re-examined, advice has to be sought, or a good attempt of examining or describing has to be made anyway. If the goal can be described another time, one should not move on to the next goal as drawn by Shepherd, but the analyzer has to go back to assess whether the goal is already carried out to a sufficient standard. I drew a dotted line in “the basic cycle of decision during task analysis” by Shepherd [Figure 6-1].

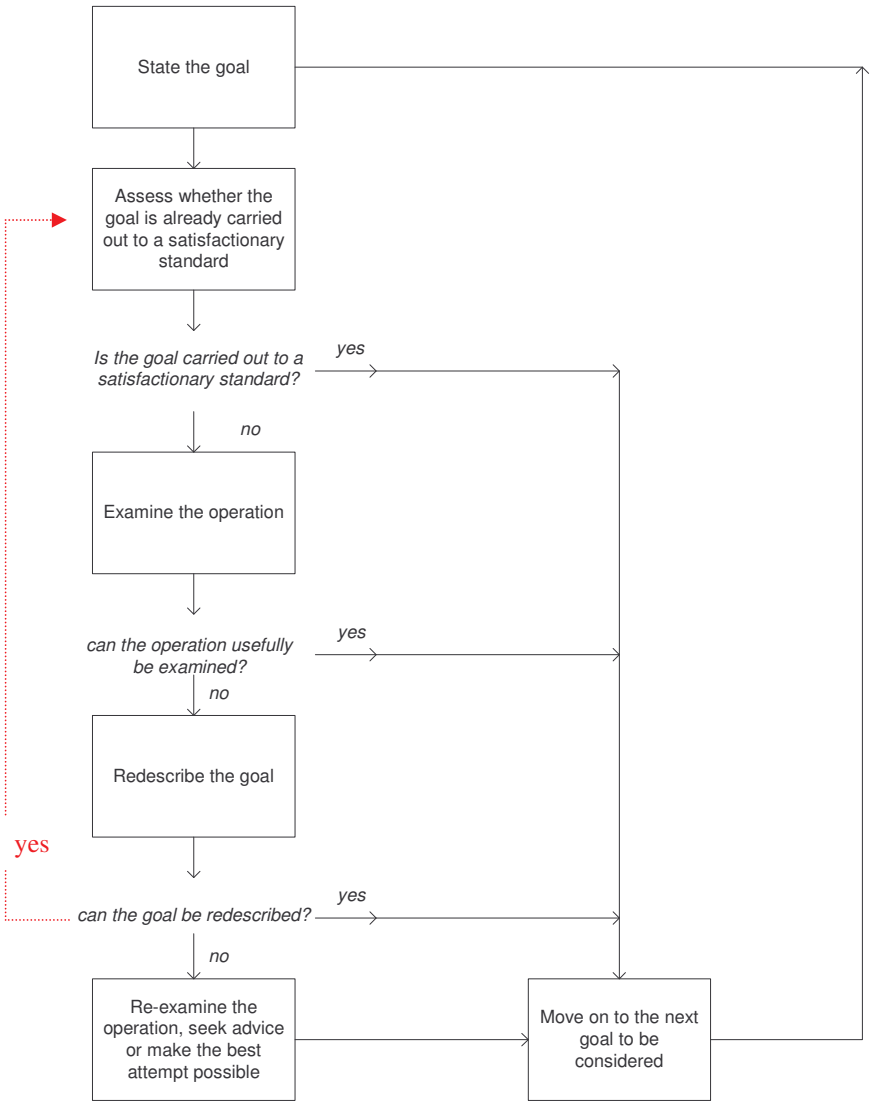


Figure 6-1 - The basic cycle of decisions during task analysis, by Shepherd [62]

6.5.2 Tasks in an HTA

Tasks are represented by a square with a number. The highest goal in the analysis is numbered with a zero. The tasks in a sequence of tasks underneath this highest goal are numbered from 1 to X, where X is the number of sequential tasks. See also Figure 6-1. The tasks are connected with lines. When an analyst wants to refer to a specific task in the diagram of tasks, this task can be referred to by the path of tasks to get there from the highest goal, with dots between them. Take for example the task at the utmost lower right of Figure 6-1. This task can be referred to as 2.3.3.

6.5.2.1 Plans

The series of task performed with the same goal is called a plan. The series of tasks belonging to the highest goal in the analysis is called plan 0. The series of tasks underneath the first task in plan 0 is called plan 1. The series of tasks underneath the third task in plan 1 is called plan 1.3, and so on. Each 'plan' has to be worked out complete. This means that all the tasks sequencing in time have to be worked out. If I find a plan of tasks interesting enough to analyze, then all of the tasks at this plan might be useful and therefore need to be analyzed.

6.5.3 Middle out HTA

A disadvantage of the HTA technique is that it works top-down. One should start with analyzing the highest goal in the task system, while I would like to start with the task of a user and find tasks as well as higher goals in order to find potentially relevant context information. The Husat Research Institute presents a way to work out the hierarchical structure of tasks in HTA, starting with a random task in the task structure [68]. Instead of working top-down, they work from the middle out. This has the disadvantage that the analysis might become less thorough as the direction where the analyst is heading is still unknown. But on the other side, the advantage is that it is possible to search for higher goals in a task analysis. One should start with a task and ask four questions about that task as shown in Figure 6-2:

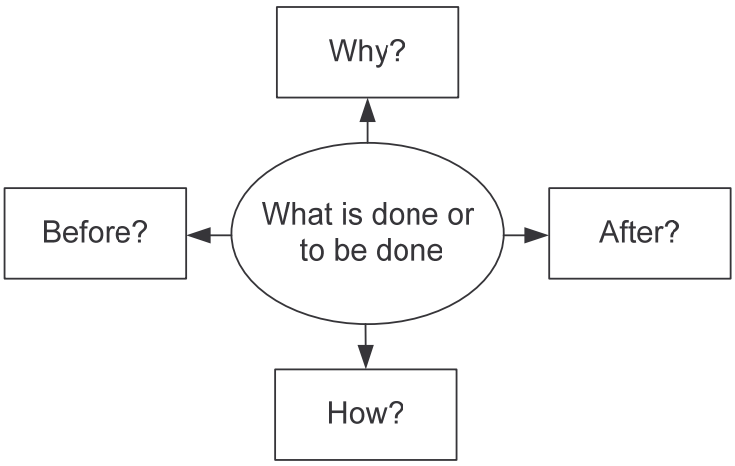


Figure 6-2 - Userfit Tools – Why? Before? After? How?

6.5.3.1 Upwards

With the question "Why?", the analyst resolves the higher goal of the described task. The answer gives the reason that the user performs this activity. The structure of goals

is chosen by the developer. In the worked example in chapter 1 I could also choose to define another reason that Anton made a call, with another higher goal. These choices can be seen as further defining the user. As I define the user by his task, and the system of tasks of this user becomes more detailed, the user is defined narrower. It is not a random decision to let goal A be the goal of the user. It is a direction the developer heads.

It is important that the goals are stated as explicit as possible. Try not to confuse goals from tasks, but try to really distinguish goals from tasks. Demands to a task can be imbedded in a task analysis, because coping with the demand will be a goal for the user. If for example the user's task is to listen to music, but that that music must have a certain sound quality, then it may be a task to listen to music, but it is the goal of the user to listen to the music with a certain sound quality.

6.5.3.2 Left and right

"Before?" is about what task is performed before the current one in order to reach the described goal. "After?" is the equivalent after the current task. These questions create a level of sequencing tasks called a 'plan' as described earlier.

Sometimes the sequence of tasks in a plan is not predetermined. If the sequence of tasks is conditional, this can be represented in a plan with if-statements or checkpoints. These conditions and checkpoints give the opportunity to describe different scenarios in one task analysis. In each plan there are more scenario's possible than can be noted. Consider for example the case that the user is disrupted in its task by some reason. This can be the result of an unlimited number of causes. An example can be found in Appendix H, plan 2.2.1. There the condition is formulated that if not Bert closes the connection, the user (Anton) will close the connection himself. Which conditions are and which conditions are not taken into account is up to the developer to decide. One criterion for him to use is the likeliness of the scenario to take place. The other is the importance of the scenario to be considered. This is analogue to the "P * C" criterion as described in the next paragraph. If the developer wants to develop context aware applications that cope with a lot of different scenarios based on the same task, he should embed a lot of conditions in his plans.

6.5.3.3 Downwards

"How?" is the question that searches for the answer on how the current task as a goal will be reached. The current task will be decomposed if possible and necessary. I will come back on this in the next paragraph.

6.5.4 The "stop-criterion" for task analyses

As mentioned in paragraph 6.5.1, in a HTA, a system of tasks is being explored in progressively greater detail until it is understood sufficiently. The task analysis therefore has to stop when a system of tasks is understood sufficiently. In HTA, the P * C rule is used to determine when to stop and when to continue in a task analysis.

*6.5.4.1 The P * C rule*

Shepherd and Kirwan and Ainsworth argue that the "P * C rule" should be used to determine whether further detail in a task analysis should be accomplished [69] [70].
P stands for probability of inadequate performance. It gives the probability that accomplishing further detail would have been valuable. P is expressed as a percentage.
C stands for Cost of inadequate performance. Here it means how important the potentially found information would be for a context aware application. C is expressed as a number. The cost of inadequate performance relate to the potential relevance of context information related with the tasks still to be explored. As a task always has the potential to give relevant context information no matter how far away from the task where this context information is relevant, there are always costs of inadequate performance.
P and C are estimated by the analyst. The product is to be calculated. A goal or task with a higher product has more reason to be investigated then one with a higher product and will therefore be investigated first.

Often, determining P and C and calculating the product will not be done explicitly by the analyzer. In a lot of cases the analyzer can make a good estimation which way to go in his task analysis. He does this by estimating whether analyzing a direction further might give important new insights. The 'whether it might' can be compared with the 'P' as described earlier and the importance of new insights can be compared with the 'C'. The "P * C rule" is used without explicitly estimating P and C in concrete numbers. An example can be seen in the worked example in chapter 1.

6.5.4.2 The lowest level

The maximum level of detail to be considered in making an HTA will be the level where the user makes physical movements. For applications this level is called the "keystroke-level". A higher detail is of no use, because the new application will not be able to support a part of a movement. The new context aware application may make entire movements unnecessary, so a whole movement can be interesting to analyze. This level of detail can also be too high. Sometimes the level where the use of an

application is described without mentioning how this application is used can be sufficient. This is up to the analyzer to decide, using the P * C rule.

6.5.5 When to stop analyzing

The ’P * C’ rule helps to choose which way to go and which way not to go. But, if the resource of time would be unlimited and a goal or task no matter how far away can potentially give relevant context information, it would be the best to keep analyzing until the end of times.

We do not have unlimited time however. There are other steps during the development of context aware applications then making a task analysis and time is money, so there are costs to consider. An analysis therefore has to stop at a certain moment. This moment is up to the ambition of the project. If the target is to develop only one context aware application, no matter with what purpose, a small task analysis with only a couple of tasks, or even just two might suffice. If however the ambition is to develop a whole series of context aware applications, or to bring multiple context aware improvements in an existing application, a larger task analysis will be useful. One could even consider doing multiple task analysis, in order to describe multiple scenarios of tasks, which each could be a source for finding relevant context information.

This time and ambition aspect could also be fitted in the ‘P * C’ rule as if a developer has the ambition to develop only a one or a few context aware applications and he thinks there is enough relevant context information to be found in the current task analysis, the importance of finding more relevant context information would be low and therefore C is low, which makes the product P * C also low.

7 Finding context aware functional requirements out of scenarios

In the previous chapter I wrote about how tasks can be found with a specific user type in mind. In this chapter I will argue how useful functional requirements can be found from a hierarchical task analysis. First I will determine when context information is relevant enough to be considered as relevant context information. Then I will present multiple ways to extract relevant context information from a worked out hierarchical task analysis.

7.1 Functional requirements

With functional requirements I mean the primary function of a future application, without thinking of usability, implementation, etc. A new found potentially useful functional requirement can later be labeled as useless if it can not be implemented in a way that the new application is useful for the user. A functional requirement can be implemented as a standalone application, or as functionality in a larger application.

7.1.1 Useful functional requirements

The methods presented in this chapter do not guarantee that ‘useful’ functional requirements will be found. They do however support the search for functional requirements which might be useful. Whether they really are useful will be determined at the moment that users actually test the application where the functional requirements are in.

7.2 The relevancy of relevant context information

In chapter 4 I argued that relevant context information relates to the user’s task. The search for relevant context information is done with the purpose of using it in an application. This application should support the user. The context information should therefore be supportive for the user in his system of tasks. It must be possible to give the support with an application on the device in question. In my case, the context information must be used in an application on a mobile device. If such an application supports the user enough for the user to actually use it, the context information was relevant enough to be considered. It is the usefulness of the application that determines the relevancy of the context information to the user.

7.3 Finding single-context aware functional requirements

In chapter 5 I set apart the difference between applications aware of one single context and applications aware of multiple contexts. I wrote that I would first look for functional requirements with awareness of a single context to combine them afterwards to find functional requirements which are aware of multiple contexts. In this paragraph I will search of functional requirements which are aware of a single context. In the next paragraph I will combine these to find functional requirements which are aware of multiple contexts.

Unfortunately there is no mathematical formula for finding potentially useful functional requirements. Cheng and Johnson wrote an article on "Applying Task Analysis to Facilitate the Design of Context-Aware Technologies" [65], but they did not give a solution for the problem how to get useful functional requirements from a task analysis. Because there are no predefined methods for finding these functional requirements based on a task analysis yet, I will introduce a couple of methods myself in order to get as much useful context aware functional requirements as possible. The functional requirement is written down as a functionality of the software, which occurs in a described context (see also paragraph 7.4.1).

In chapter 4 I presented the three advantages of context aware applications, as stated by Mari Korkea-aho [58]. Useful applications and with that, relevant context information, can be found by analyzing the task analysis to find applications with one of the three context aware advantage characteristics. I will now present three methods to do that. In each method eventually the concrete functional requirements will be formulated. Each of these functional requirements will get a code of the scenario combined with a number. B25 for example stands for functional requirement number 25 of scenario B.

7.3.1 Method 1: Information and services presented to the user according to the user's context

The first possible advantage of context awareness can be split in two. One is about presenting information and the other one is about presenting services according to the user's context. Both should be used to find relating useful context aware functional requirements. I start with the services. Services can be presented to the user according to the context of the user. Useful context aware functional requirements can be found when you:

Recognize the needed services in the task analysis and analyze in what context the services are used.

This should not five any problems. The information part is more difficult. The user may need information for making decisions in his task, or finding information can be a part of his task.

Functional requirements can be found by analyzing the information-need for all decisions that are made in the task analysis.

The developer should place himself in the position of the user and wonder what questions he would ask himself to get the information needed for the decision. Out of these questions functional requirements of context aware applications should be formulated. Not all information needed for the decisions will require context awareness. The developer should decide whether he wants to implement this non context aware functionality in his future application or not. As this is a creative task for the developer, it might be useful to let other people help in this step as people inspire each other creatively.

I will now discuss some issues regarding this method before continuing with the next.

7.3.1.1 Not all decision need extra information

A lot of decisions normally do not need a lot of information. An example is task 2.3.1.1: 'Anton creates a connection' in the worked example in chapter 1. He creates this connection to make a call. He decides to do it, but he does not have a lot of choice. If he wants to call somebody, he will need to create a connection. This decision can be made without further information and is therefore not interesting for our purposes.

7.3.1.2 Generic services in different contexts

The user's services need might depend on his context. In the task analysis the needed services should be recognized and the context where they are necessary should be analyzed. Applications can present the maybe even generic services according to the described context.

7.3.1.3 Coping with new found goals

The tasks described in the task analysis have the goal that is stated in the same task analysis. But with decision being made also other goals are in consideration. Take for example task 2.2.1.1.1: 'Anton grabs his mobile'. He could also grab his fixed line

telephone. This decision is not based on the goal to create a connection, because both types of phone could be used to make a connection. The decision does also not have a relation with one of the higher goals in the described hierarchy. One reason to take his mobile phone can for example be that using this phone saves time because should have walked to another room. These other reasons can be found by the analyzer by transferring himself in the place of the user and wondering what would be important for him making his decision. The analyzer could have a look at Maslov's theory of basic needs for inspiration [appendix **Error! Reference source not found.**]. There are two ways to cope with the problem of finding new goals.

7.3.1.3.1 Updating the task analysis when finding new goals at decisions

At first the analyst could update his task analysis. The goal then will not be to create a connection, but to create a connection in as less time as possible. But then also goals higher in the hierarchy have to be updated. And not only with this goal, but also with other goals that may pop up at making the decision to make a connection with his mobile phone. He should not only do that for this decision, but for all the decisions being made. This method makes the process of task analyzing an iterative process, which makes the task analysis more complex, but also more complete. It is up to the analyzer to decide which he prefers.

7.3.1.3.2 Keeping secondary goals out of the task analysis

An alternative is that only the primary goals are mentioned in the task analysis and that other, secondary goals are described only at the points where decisions are being made. And these goals are used this way to find context aware applications. The advantage of this method is that the task analysis does not get more complex, but the process of finding context aware applications will be more arbitrary because the analyzer comes up with goals that are not defined in the task analysis.

7.3.2 Method 2: Automatic execution of a service when in a certain context

The automation of tasks is an old application of ICT. In the early days mostly administrative tasks like overtyping of documents were automated. Context information can help replacing explicit user input by implicit input in order to automate tasks or automatic execute a service [65] [71] [72].

> *One should wonder what tasks could be automated and what context aware functional requirements are needed for that.*

7.3.3 Method 3: Tagging of context to information for later retrieval

Tagging of information has an advantage at retrieval.

> *One should look where information is used in a task analysis and whether it would have been useful that it was tagged.*

Like method one, this is a creative task for the developer. It might therefore be useful to let other people help in this step as people inspire each other creatively. Also with this method functional requirements should be formulated.

7.4 Finding generic context aware functional requirements

In the previous paragraph I described how to find useful functional requirements from a scenario. The functional requirement describes how an application should behave in a certain context. There is a relation between the single task that is supported and the context that can be found somewhere higher in the task analysis [Figure 7-1]. Take for example the task that Anton chooses Anna in the phonebook of his phone. The phone could have "sensed" that Anton sent Anna a proposal. It could have used this information to suggest Anton to call Anna instead of the current situation where Anton has to find Anna in his phonebook the same way as any other person in his phonebook.

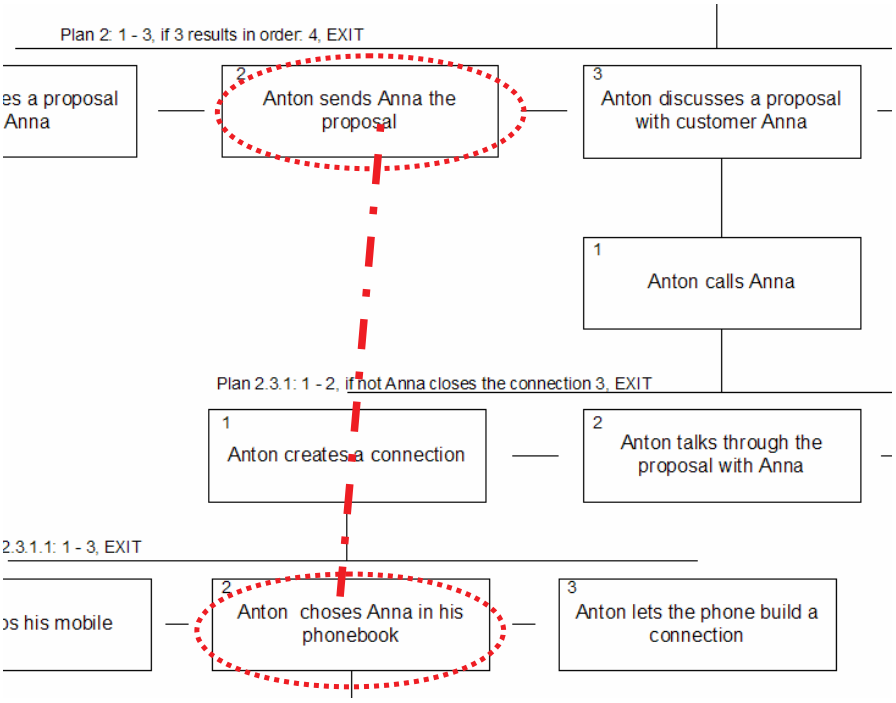


Figure 7-1 - The context of Anton choosing Anna in his phonebook

In this paragraph I will set apart how to find generic context aware functional requirements. This will result in that not only a single context is found for a functional requirement, but that multiple contexts are found which all trigger about the same behavior. The example where a user has to choose someone to call can for example be extended with occurrences in scenario B. The context of Anton choosing Bert is illustrated in Figure 7-2. The different contexts, in which a generic functional requirement can occur, are illustrated in Figure 7-3.

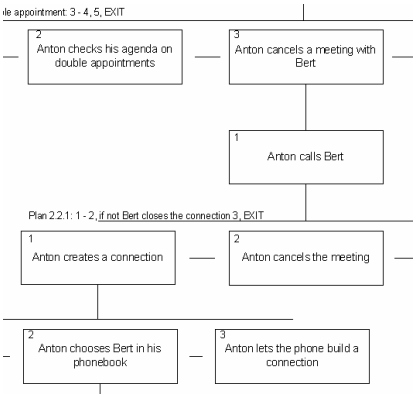


Figure 7-2 – Anton chooses Bert

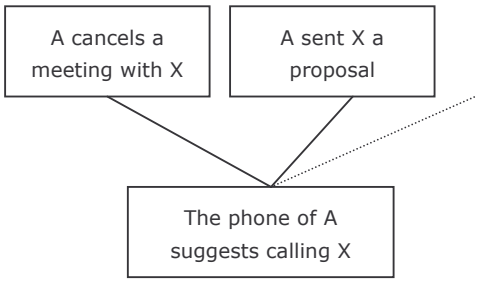


Figure 7-3 – The phone of A suggests calling X

How these generic context aware applications can be recognized, is described in the following paragraph: 7.4.1.

7.4.1 Recognizing generic context aware applications from single-context aware functional requirements

In paragraph 5.2 I described that I would find relevant context by finding functional requirements from task analysis. In this paragraph I will describe how to find generic context aware functional requirements whose functionalities differ in different contexts.

In the worked example in paragraph 10.4 I describe two newly found single-context aware functional requirements, A22 and B26, as two instances of the same generic context aware functionality. In the current paragraph I will explain how generic context aware functional requirements can be found out of single-context aware functional requirements

1. First, the example names in the found functional requirements should be replaced by more abstract descriptions they already represented anyway.

2. Second, the found single context aware functionalities have to be split up in:
    1. the behavior
    2. the context in which this behavior is planned to occur
3. Third, through all the single context aware functional requirements of all the different scenarios, the functionalities where the same behaviors are described have to be put together.

    Figure 7-3 shows how one behavior can result from multiple contexts.
4. Fourth, the contexts in which the behavior occurs have to be generalized.
5. Finally, the resulting generic context aware functional requirement has to be described.

7.5 All context aware functional requirements found

After combining a list of context aware functional requirements found in the different scenarios, the list of functional requirements found in the scenarios and the generic context aware functional requirements found by combining these functional requirements are all available to choose from for further application development.

8 From context aware functional requirements to applications

The last step in the method is to make real working applications out of the suggested concepts. From a SDM point of view, like described in paragraph 5.1, the information planning stage has to be finished before continuing to the next stage of development. I have no indication that further application development out of the found primary functional requirements will differ from developing regular applications.

The suggested functional requirements have to be tested on things like for example technical feasibility, expected profitability and privacy laws. Then a choice has to be made whether one, some, or all functional requirements will be build to a prototype. This prototype can be used for testing whether the user will really like the functional requirement and if he likes it, the prototype can be used to optimize the user interface.

Also other aspects of a project planning have to be considered. It is out of the scope of this thesis to specify these tasks further, but for developing applications they are to be done.

9 Finding context aware functional requirements in a nutshell

In this chapter I will summarize the steps from user to context aware applications. For each the paragraph with more details of that specific step is stated.

1 Define the user by his tasks, define the group he belongs to and set down assumptions that are made (paragraph 6.2).

2 Make a hierarchical task analysis using the middle out technique and the 'P * C' and key level stop criteria for all scenarios (paragraph 6.5).

3 Analyze the information-need for all decisions that are made in the task analysis and analyze how information retrieval as a task itself can be supported (paragraph 7.3.1).

4 Recognize the needed services in the task analysis and analyze in what context the services are used (paragraph 7.3.1).

5 Wonder what tasks could be automated and what context information is needed for that (paragraph 7.3.2).

6 Look where information is used in a task analysis and whether it would have been useful that it was tagged (paragraph 7.3.3).

7 Repeat the steps 3 till 6 for all scenarios.

8 Replace the names in the found functional requirements by abstract names and split them up in a behavior and the context in which this behavior occurs. Then look for descriptions of the same behavior in different contexts. Finally, the contexts in which the behavior occurs should be generalized and the generic context aware functional requirement has to be described (paragraph 7.4).

9 Use an existing method to test and build context aware applications out of the found functional requirements (chapter 8).

Part 3 Validation

10 Validation by use of an example

In this chapter I will validate the method of finding context aware functional requirements by working out an example. In the heading of each paragraph the steps that are described are named. The example is used to show that the method can indeed be used to find context aware functional requirements. Parts of it will also function as illustration to writings earlier in the report.

10.1 Define the user by his tasks, define the group he belongs to, define the scenarios he is in and set down assumptions that are made (step 1)

Because I did this research in an organization where a lot of mobile applications are developed for mobile operators, I choose to support a user that makes a phone call. This is not a very exotic, modern, or exciting task. But by using such an “every day”-task, I want to show that the method can be used also for finding context aware functional requirements to support users with such a task. I named the user Anton, but an abstract user ‘A’ can be read where ‘Anton’ is written. In order to narrow the user a bit I define him as:

The user is a business man calling someone.

The scenarios are that the business man calls with changing goals. In each scenario another person appears, namely the person who is called for the changing reason. Here also real names (Anna, Bert, Chris and Dirk) stand for abstract persons (A, B, C and D). I chose to use the following four scenarios:

A Discuss a proposal
B Cancel a meeting
C Order a taxi
D Work out a problem

Further assumptions will be made further in the method.

10.2 Make a hierarchical task analysis using the middle out technique and the 'P * C' and key level stop criteria (step 2)

The task analysis diagram I made for the scenario where the businessman calls in order to cancel a meeting is added as appendix **G** and worked out below. The other diagrams for the other scenarios are added as appendix D, E and G. I gave the user a name (Anton) to make the example more vivid. Other persons in the task analysis also received names.

1 I started writing down and drawing the task of Anton, namely calling Bert.
2 For this task I wondered what happened before and after Anton's 'prime' task, why this is his task and how he did it. As this is only the beginning of the analysis, I presume all findings to be relevant.
3 Downwards, I continue to expand the task analysis until the keystroke level is described. I considered whether these newfound tasks would potentially be useful for finding relevant context information and concluded they would. The probability that relevant information could be extracted out of these tasks is high. The value of the information to be found is average to high.

Plan 2.2.1.1.2 is an example of a sequence of tasks that were worked out until and including the keystroke level, presuming that opening a phonebook can be done with one touch of a button, which I see as one movement.

4 Upwards, I continue expanding the diagram until I received the highest goal I want to have in the task analysis, namely that the user manages his agenda. I stop developing further because I think that quit some relevant context information can be found in this task structure and that is enough for my ambition with this example.

As I already met that objective, the value of more context information to be found is low and therefore the cost for not finding that context information (C) is also low. For that reason, P * C will be low for the goal above the current higher goal and therefore I will not expand the task analysis that way.

5 Each plan is worked out completely. In plan 2 I add one condition. Namely that double appointments are found. If they are found, then a meeting has to be cancelled and the meeting has to be removed from the agenda before the agenda can be closed again.

In plan 2.3.1 I added the condition that it is not Bert who closes the connection. If Bert does close the connection, Anton does not have to do that anymore.

10.3 Find useful context aware functional requirements (step 3 – 7)

Useful functional requirements are found for all the scenarios. These scenarios are worked out in appendices G, L, M and N for scenarios A, B, C and D. The scenarios A and B are fully worked out including the functional requirements. Scenarios C and D are worked out until the description of functional requirements due to time restrictions.

10.4 Find generic context aware functional requirements (step 8)

Replace the names in the found functional requirements by abstract names and split them up in a behavior and the context in which this behavior occurs. Then look for descriptions of the same behavior in different contexts. Finally, the contexts in which the behavior occurs should be generalized and the generic context aware functional requirement has to be described (step 8)

After replacing the names with abstract ones and splitting up behavior and context, I found for example the behavior that the phone suggests calling someone based on the contexts.

In order to make clear what tasks were used further on in this paragraph, I will show them here explicitly. Normally all the single context aware functional requirements have to be split and rewritten before the matching behaviors will be found. The two following tasks where used:

2.3.1.1.2 Anton chooses Anna in the phonebook

> *A22 As Anton selected Anna to be called as a renewed customer, his phone could have suggested calling Anna on a prominent place on the interface of his mobile phone.*

2.3.1.1.2 Anton chooses Bert in his phonebook

> *B26 As Anton was busy in his agenda with the meeting with Bert, his phone could have suggested calling Bert on a prominent place on the interface of his mobile phone.*

They were rewritten to:

> *A22 behavior: The phone of A could have suggested calling X on a prominent place on the interface of his mobile phone*

A22 context: As A selected X to be called as a renewed customer

B26 behavior: The phone of A could have suggested calling X on a prominent place on the interface of his mobile phone
B26 context: As A was busy in his agenda with the meeting with X

These rewritten functional requirements can be illustrated in the following figure.

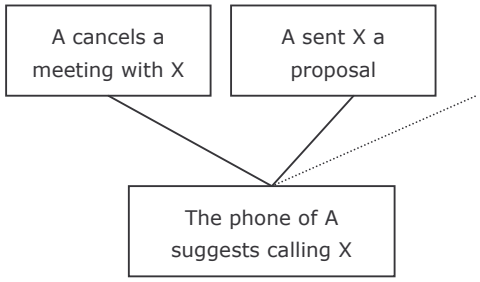


Figure 10-1 - The phone of A suggests calling X

Finally, the contexts in which the behavior occurs should be generalized. In this case I generalize the contexts with:

"a person occurring in applications used by the user".

The generic context aware functional requirement resulting from the steps above is:

"A phone could suggest his user to call a certain person when this person occurs in an application the user is using."

10.5 Building and testing context aware applications (step 9)

From the functional requirements found in the last paragraph, a selection has to be made using an existing method before these functional requirements are the target primary functional requirements of the future context aware application. As this is not a part of the method, the worked example stops here.

11 Evaluation of the worked example

In the previous chapter an example was worked out using the method presented in this thesis. Here I will evaluate how the method worked in this example.

11.1 Improvements out of the worked example

Using the method in an example forced me to use the method step by step so that I noticed where in the new method errors occurred and additional information or additional steps where necessary. It also made it easier to discuss the use in practice of the method with people from TNO. Not the content of the example was interesting, but the way the method behaved in practice was.

The most occurring problem was, that too less information was added to some steps in the method, to do these steps the way that they were meant to be done. This problem and some minor other are all fixed by iteratively improving the descriptions of the steps.

11.2 Recommendations for further development

At certain points in the method, some intelligence or creativity is demanded from the developer. These tasks for the developer could be supported.

11.2.1 Further guiding the developer through hard developer tasks

In step three of the method for example, the developer is asked to recognize the decision made in the task analysis and then put him self in place of the user and wonder what information he might need to support his decision. It might be useful to include a more concrete method for recognizing the decisions.

11.2.2 Support of creative developer tasks

It also might be useful to have support in the creative process of thinking what information might support these decisions. From this information especially context information is interesting as the developer is looking for context aware functional requirements.

11.2.3 A list of context information to tag to information for easier later retrieval

Finding out whether it is useful to tag information with context information for easier later retrieval, could be supported by a list of possibly useful context for later information retrieval.

11.2.4 Improving the method further by experiences in a business environment

The method is evaluated by working out an example. This example however is not worked out in a business environment. It might be useful to improve the method further using the experiences with it in a business environment.

11.2.5 Validating the method by finding context aware functional requirements for other than mobile applications

The method is developed as a generic method for finding context aware functional requirements for applications. It is validated only for finding context aware functional requirements for mobile applications. Although there are no reasons to believe that the method will not work for finding other than mobile applications, this should still be verified by using it for other than mobile applications.

12 Finding context aware versus finding regular functional requirements

In this thesis I present a new method. Not to develop regular functional requirements, but to develop context aware ones. If you are a developer, should you use the method presented in this research, or should you keep using your favorite development method? Well, that depends.

First you should wonder whether you want to build context aware applications or you just want to build an application. In this research I do not have the ambition to change the world of application development drastically and convince every developer to from now on only develop context aware applications. I do not even want to make a fair comparison between regular applications and context aware applications. All I do is provide an introduction in the world of context awareness before presenting a method on how to develop context aware functional requirements. I mention the advantages of context awareness and describe how to find functional requirements that have these advantages. It is up to the developer to decide whether he wants to build context aware applications, but if he wants to build them, he should consider using this method for finding context aware functional requirements. If he does not want to develop context aware applications however, I would recommend to keep making use of an existing application development method which might include a method to find functional requirements as the method I present is only aimed at finding context aware functional requirements.

12.1 The main differences between the processes of finding context aware and regular functional requirements

If the developer considers using the presented method for developing context aware functional requirements, the adaptation from a conventional method will not be very big. Presuming that the developer is used to developing applications from a user centered point of view, there are only a few differences in this technique in comparison with regular user centered functional requirement finding.

12.1.1 From user to tasks

In the step from user to task, the hierarchical task analysis is used. This technique is already broadly used for developing applications [84] [85] and will therefore not be

hard to use by a developer. There is a small adaptation used in the method which makes it possible to start analyzing with the user's primary task and work up- as well as downwards through his hierarchy of tasks instead of working top down. The stop criteria differ from regular applications not by method, but by choices. The standard HTA 'P * C' method is used to choose what is taken into account and what is not. For context aware applications normally more tasks are analyzed and analyzing upwards, towards "higher" goals, is possible.

12.1.2 Finding useful context aware functional requirements

The methods for finding useful context aware functional requirements are totally new. These concepts are specially designed for the purpose of finding context aware functional requirements and a developer will therefore not have any experience with using these methods. They are however not very complicated.

12.1.3 Building and testing context aware applications

There is no special method for building and testing context aware applications out of context aware functional requirements. The method prescribes that regular methods should be used for this step because I recognized no special demands for context aware applications.

13 Finding functional requirements in a structured or an ad-hoc way

In chapter 4 I described some existing context aware applications. In chapter 1 I stated that until now, there were no methods for finding context aware functional requirements. This means that a current context aware application:

- was developed without the intention to make it context aware, but it accidentally is ,maybe without even recognizing it as so.
- or it was developed using a self-developed method that was not published afterwards.
- or it was developed more or less ad-hoc.

Research is needed to get exact information on how the functional requirements of these applications were found, but in each of these cases no existing structured method was used. Developers who do have the intention to develop context aware applications now have the choice to find context aware functional requirements with or without the support of a method which structures the process.

I would urge these developers to use a method which structures the process. If for some reason you do not like the method presented in this research, then try to adapt it to your needs or develop something on your own. The advantages of using a method like the one I presented in this research are:

- It forces the developer and with that helps him to strictly define his target user, which is important to find functional requirements that actually support the user.
- It helps the developer to think about what tasks and goals a user has in order to decide which of these he wants to support with the new application.
- It helps the developer to try and find context aware functional requirements for all the tasks he wants to support and with all the potential context aware advantages that are known.

Part 4 The final words

14 Conclusions and recommendations

In this chapter I will write a conclusion by answering the research question and I will give recommendations for further research.

14.1 Conclusion

In the introduction of this thesis I asked myself the following research question:

How can useful context aware functional requirements be found?

The answer is that useful context aware functional requirements can be found by using the method presented in this thesis.

In that same introduction I stated some sub questions. The answers to those sub questions together form an answer to the main research question. I noted the most important conclusions from each part of the thesis below.

14.1.1 Conclusions on context awareness (part 1 of this report)

Dey's definitions of context and context awareness appeared very useful in finding context aware functional requirements.

> *"Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves (…) where relevancy depends on the user's task."*

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

Three possible advantages of context awareness defined by Mari Korkea-aho will be useful in the development of the method:

1. Information and services can be presented to the user according to the current context.
2. Automatic execution of a service when in a certain context
3. Tagging of context to information for later retrieval

14.1.2 Conclusions on the method (part 2)

As no method for finding functional requirements already exists, I decided to develop one myself. The definitions of Dey make clear that the relevancy of context information depends on the user's task. It is therefore not only possible to find context aware functional requirements starting with a user, but it is essential. The tasks are defined by multiple task analysis, called scenarios, using an adaptation of the hierarchical task analysis method, which supports finding goals of tasks. In each scenario functional requirements are found from the task analysis by finding out how the three possible advantages of context awareness as defined by Mari Korkea-aho can support these tasks.

The output for each scenario is a list of potentially useful functional requirements. Scenarios increase the potential usefulness of found functional requirements in different situations. The found functional requirements from the different scenarios are compared. The functional requirements which describe more or less the same functionality are used to describe a generic context aware functional requirement, which might not only be useful for the instances it originates from, but also in other instances yet to be discovered. Functional requirements that do not appear in different scenarios might also be useful in different scenarios, but as their usefulness is only validated for one scenario, that is not sure.

Due to time restrictions, unfortunately it was not possible to include a method for selecting the most useful functional requirements. When the functional requirements which are to be developed further are selected, regular application development methods can be used to develop working context aware applications out of these functional requirements.

14.1.3 Conclusions on the validation (part 3)

The validation by working out an example made clear where the method needed some more attention, but also that the method was useful for finding context aware functional requirements. Although these shortcomings where all repaired, the method might still have some small shortcomings when used practice, because no 'real' applications have been developed while using this method yet. Another issue is that the model was only validated for mobile applications.

14.2 Recommendation 1: The next step in the method

No ranking for usefulness is made in this method. The method delivers more functional requirements than will still be useful implemented, than are technically feasible, commercially attractive, or than the developer wants to or even is able to use in one or more context aware application(s). It would therefore be useful to extend the method with a step to select the functional requirements that will actually be developed.

14.3 Recommendation 2: More support in the method

In the evaluation of the worked example it became clear that it might be useful to further guide the developer through tough developer tasks, to further support creative developer tasks and to introduce a list of context information that might be interesting for tagging to information for later retrieval.

14.4 Recommendation 3: Further evaluation

The method is evaluated by working out an example. This example however is not worked out in a business environment. It might be useful to improve the method further using the experiences with it in a business environment. It would also be useful to validate the method for finding other than mobile applications.

15 Reflection

In this stage I will have a moment of reflection on this thesis. What have I done, how did it go and what have I accomplished with this thesis.

15.1 Scientific relevance

In this thesis I presented a method for finding context aware functional requirements for future context aware applications. I combined existing theories on context awareness and used them to develop a method which makes it possible for a developer to find potential useful context aware functional requirements for applications which support a user. Context awareness is written about for more than ten years and context aware applications have been developed even longer without the meaning to build context aware applications. But for the last ten years applications are being developed with the purpose to make them context aware. Until now it seemed that developers did not develop context aware applications in a structured way. They may have secretly used a structured method but did not publish it, they managed to develop context aware applications without a method to support them and that went well, or they did not have a choice because no method was available.

As I could not find any method to develop context aware applications I presume that no method was available yet. By introducing this method, now there is. Although it does not cover the entire process of application development from beginning till the end, it does support the difficult part of finding a primary functional requirement for context aware applications. Regular non-context aware methods can be used for further developing these functional requirements to working applications.

15.2 Practical relevance

One organization that develops context aware applications is the organization where I did this research, TNO. It was in cooperation with employees of TNO that this method was developed. While I am writing this reflection, the method is being used in one of their projects. Although the time is missing to fully evaluate this practical case, the responses of the users of this method are still only positive. The structured way of finding context aware functional requirements forces to explicitly define things that where previously seen different by the different project team members but are only now

discovered using this method. No more can be said at the moment because the project has only just begun.

15.3 The process of writing the thesis

Writing a thesis is a process with a lot of developing insight. This results in a lot of changes in subject. The context awareness part has been the constant factor and also ‘developing’ has never left this thesis. But I started with the intention to develop one or more context aware applications and I finished with developing a method that covers the first part of developing these context aware applications. Not because I do not like to program software, I do, but because I could not find a suitable method to determine what context aware application I would like to develop. As I want to explain all the choices I make, I did not want to develop a random application. The need to explain all the choices I make is still in me. I think a lot of choices made in this research are explained, but I know that not all are. I struggled to explain all choices during the last months. I wanted to explain it all, but time is running out. ‘Sufficient’ explanation is the next best thing and I think I sufficiently explained my choices. This struggle however did not give me a hard time. The struggle was just a motivation to deliver something that can be put in practice. Unlike a lot of fellow students I did not have a hard period somewhere in my research. I liked the subject, I liked to work on it and I liked to discuss it with my supervisors. I might even try to continue researching on the subject in the future.

References

[1] Annual Report Nokia 2004, http://www.nokia.com

[2] http://www.nokia.com/nokia/0,,73210,00.html

[3] http://logistics.about.com/library/weekly/uc110102a.htm

[4] http://dictionary.reference.com/search?q=adapt

[5] http://wap.sonyericsson.com

[6] http://www.webopedia.com/TERM/W/WAP.html

[7] http://wap.yahoo.com

[8] http://www.mobile-phones-uk.org.uk/sony-ericsson-p910i.htm

[9] http://www.abc.net.au/widebay/stories/s1428825.htm

[10] http://www.slashphone.com/83/1719.html

[11] http://www.gsmworld.com/news/press_2005/press05_21.shtml

[12] http://www.gsmworld.com/using/billing/potential.shtml

[13] Trosby, F., SMS, the strange duckling of GSM, Telenor Nordic Mobile, 2004

[14] http://www.answers.com/topic/short-message-service

[15] http://www.gsmworld.com/technology/gprs/intro.shtml

[16] http://www.darwinmag.com/learn/curve/column.html?ArticleID=182

[17] 3G Break Even Doubtful, Forrester, 7th October 2002

[18] EDGE, Introduction of high-speed data in GSM/GPRS networks, Ericsson, 2003

[19] Maslov, A.H., A Theory of Human Motivation, Psychological Review, 1943

[20] Australian Unity Wellbeing Index, Survey 3, Report 3.1, May 2002

[21] InPsych Highlights, P.Bradley, February 2003

[22] Boeree, Dr. C.G., ABRAHAM MASLOW

[23] Markus, H., Kitayama, S., The what, why, and how of cultural psychology, 1992

[24] Maslov, A.H., Motivation and Personality, 2nd ed, 1970

[25] Coe, D.F., Maslow's Marketing Filter, 2004

[26] http://www.nokia.com/5140

[27] http://www.irda.org

[28] Suvak, D., IrDA and Bluetooth: A Complementary Comparison, 2002

[29] http://en.wikipedia.org/wiki/WLAN

[30] Stone, B., Mobile Phones Could Make Your PC Obsolete, Newsweek, April 2005

[31] http://www.nokia.com/3510

[32] http://phonearena.com/htmls/readreviews.php?id=557&page=4&category=

[33] http://www.nokia.com/6100

[34] http://dictionary.reference.com/search?q=context

[35] www.une.edu.au/ltc/Plagiarism/glossary.htm

[36] http://www.cs.helsinki.fi/u/kraatika/Courses/f4fMC/ContextAwareness.pdf

[37] Schilit, B., Adams, N., Want, R., Context-Aware Computing Applications, Xerox Corporation, 1994

[38] Wegdam, M., e.a., An overview of the AWARENESS project, Freeband, December 2004

[39] Schilit, B.N., Theimer, M.M., Disseminating Active Map Information to Mobile Hosts, Columbia University, 1994

[40] Pasco, J., Adding generic contextual capabilities to wearable computers, University of Kent at Canterbury, 1998

[41] Dey, A.K., Understanding and Using Context, Georgia Institute of Technology, 2001

[42] Riva, O., A Conceptual Model for Structuring Context-Aware Applications, University of Helsinky, 2004.

[43] http://www.cs.helsinki.fi/u/kraatika/Courses/f4fMC/ContextAwareness.pdf, University of Helsinky, 20th September 2004

[44] Sadjadi, S.M., Supporting Programming Paradigms for Developing Adaptive Applications, Michigan State University, October 2002

[45] http://www.microsoft.com/windowsmobile/default.mspx

[46] http://www.symbian.com

[47] Dey, A.K., Abowd, G.D., Toward a better understanding of context and context-awareness, Georgia Institute of Technology, 1999

[48] Pascoe, J., Ryan, N.S., Morse, D.R., Issues in Developing Context-Aware Computing, Springer-Verlag, 1999

[49] Schilit, B., Adams, N., Want, R., Context-Aware Computing Applications, Xerox coorporation, 1994

[50] http://www.nokia.com/nokia/0,,48546,00.html

[51] http://www.nokia.com/nokia/0,,48542,00.html

[52] Siewiorek, D., e.a., SenSay: A Context-Aware Mobile Phone, Carnegie Mellon University, 2002

[53] http://www.colorado.edu/geography/gcraft/notes/gps/gps_f.html

[54] http://www.enginyoy.com

[55] http://www.66.com

[56] http://www.loc8tion.com

[57] http://www.nokia.com/nokia/0,1522,,00.html?orig=/sensor

[58] Korkea-aho, M., Context-Aware Applications Survey, Department of Computer Science Helsinki University of Technology, April 25 2000

[59] http://www.tno.nl

[60] Cooper, A., Reiman, R., About Face 2.0, The Essentials of Interaction design, Wiley Publishing. Inc., Indianapolis, 2003, p.12

[61] Norman, D.A., The Design of Everyday Things, Basic Books, New York, 1988

[62] Shepherd, A., Taylor, Francis, Hierarchical Task Analysis, London, 2001, p. 26

[63] Cooper, A., About Face, The Essentials of User Interface Design, IDG Books Worldwide,Inc., California, 1995, p. 555-556

[64] Kirwan, B., Ainsworth, L.K., Taylor, Francis A guide to task analysis, London, 1992, p. 81-145

[65] Cheng, Y.M., Johnson, C., Applying Task Analysis to Facilite the design of Context-Aware Technologies, Academia Sinica, 2002

[66] Cooper, A., Reiman, R., About Face 2.0, The Essentials of Interaction design, Wiley Publishing. Inc., Indianapolis, 2003, p.45-53

[67] Annett, J., Duncan, K.D., Task analysis and training design, Journal of Occupational Psychology, 41, 1967, p.211-221.

[68] Task analysis, Tools & Techniques, Userfit Tools, 1994

[69] Shepherd, A., Taylor, Francis, Hierarchical Task Analysis, London, 2001, p. 27

[70] Kirwan, B., Ainsworth, L.K., Taylor, Francis A guide to task analysis, London, 1992, p. 109-111

[71] Selker, T., Burleson, W., Context-aware design and interaction in computer systems, 2000

[72] Dey, A.k., Abowd, G.D., The Context Toolkit: Aiding the Development of Context-Aware Applications, 1999

[73] Luo, L., John, B.E., Predicting Task Execution Time on Handheld Devices Using the Keystroke-Level Model, Carnegie Mellon University, 2004

[74] Kleinbauer, T., Bauer, M., Jameson, A., SPECTER: a User-Centered View on Ubiquitous Computing, German Research Center for Artificial Intelligence, 2002

[75] TNO ICT intranet

[76] http://www.freeband.nl/project.cfm?id=530

[77] http://www.freeband.nl/project.cfm?id=528

[78] Paterno, F., Task Models in Interactive Software Systems, CNUCE—C.N.R, 2000

[79] Pahl, G., Beitz, W., Engineering Design, A systematic approach, 1996, London, p 140-149

[80] http://www.slougheton.slough.sch.uk/classes/ict/ks3/glossary/

[81] http://www.ssuet.edu.pk/taimoor/books/0-7897-1063-3/appa.htm

[82] http://www.uta.edu/infosys/e_comm/terms/term_a.htm

[83] http://www.tcnj.edu/~cobb2/Glossary.htm

[84] Hoffman, R.R., Elm, W.C., Handcuffs on the Development of Human-Centered Information Technology, Florida Institute for Human and Machine Cognition, 2005

[85] Zhang, P., Integrating Human-Computer interaction development into the systems development life cycle: a methodology, Syracuse University, 2005

[86] http://appl.nasa.gov/resources/lexicon/terms_f.html

[87] http://lookup.computerlanguage.com/host_app/search

[88] Lemmen, K.A.M., Methodologie van informatiesysteem ontwikkeling, Open Universiteit, Heerlen, 1993

Acronyms

BU	Business Unit
CA	Context Aware
HTA	Hierarchical Task Analysis
GOMS	Goals, Operators, Methods, and Selection Rules
GPS	Global Positioning System
GSM	Global System for Mobile Communications
LAN	Local Area Network
SDM	System Development Methodology
TA	Task Analysis
TNO	Toegepast Natuurwetenschappelijk Onderzoek. In English: 'the Netherlands Organisation for Applied Scientific Research'
TNO ICT	TNO Information and Communication Technology
UMTS	Universal Mobile Telecommunications System
USB	Universal Serial Bus
UCI	(department of) User Centered Innovation
WLAN	Wireless LAN

A Basic Needs

The relevancy of context is related to the user's task. The task of the user is represented in this method as a hierarchy of task, with upwards the goal of each task, being a task for a higher itself. This process of finding the goal of a goal can not be continued unlimitedly. Maslow wrote a theory on the basic needs of human beings. These basic needs can be regarded as the ultimate user goals. In this appendix I set down Maslow's theory of human motivation about basic human needs.

A.1 What basic needs do people have?

It is interesting to find out whether people have some basic needs. A product that helps an individual fulfill these basic needs has a competitive advantage. In 1943 Abraham Maslow introduced the paper that made him famous: "A theory of human motivation" [19]. In this paper he introduced a hierarchy of human basic needs. One need must at least partly be fulfilled before the next one gains importance.

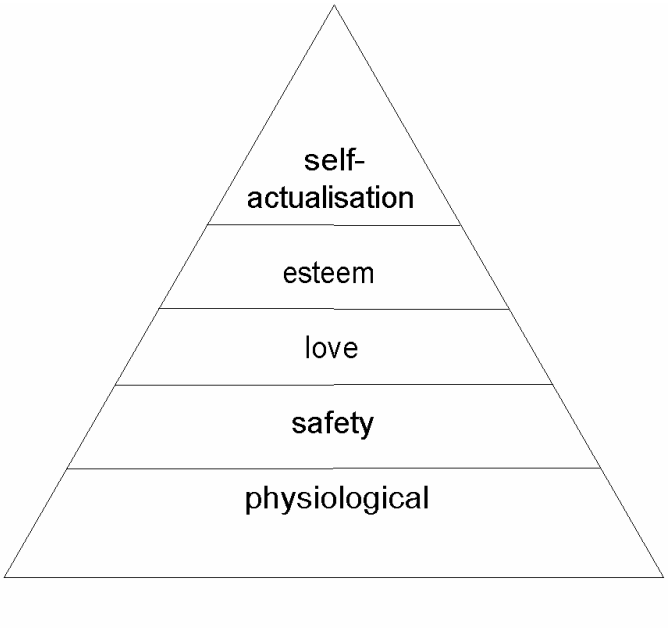


Figure 0-1 Maslow's pyramid of needs

A.1.1 Physiological

This hierarchy can be shown in a pyramid with the most basic need, the physiological needs as a basis. Maslow for example recognized sleep and sex as physiological needs. Maslow saw great importance in the concept of homeostasis. Homeostasis refers to the

body's automatic efforts to maintain a constant, normal state of the blood stream. Therefore it is important to be able to get the right amount of water, sugar, protein, oxygen, warmth, etc. If someone is really hungry, all he wants is food. Nothing else is important. The less someone is hungry, the more important is the second need: safety.

A.1.2 Safety

In a survey conducted after the September 11 terrorist attacks, 90% of the Australians surveyed felt sadder as a result of the attacks, with the great majority rating their distress at a very high level [20]. These people felt that their safety was not what it used to be. Therefore they supported the use of violence against the people who masterminded the attacks [21]. If the physiological needs are largely fulfilled and people feel unsafe, they will do anything to feel safe again.

A.1.3 Love

The third basic need is the need for love. If one feels well physiologically and also save, he will search for love. Love means giving as well as receiving. Love is not sex.. Maslow sees sex as a physiological need. Love is not only with a lover, but also with friends and family. Love is about the people around one, about social acceptance.

A.1.4 Esteem

Next in the row is esteem. People have a need for a firmly based, stable, high evaluation of themselves, for self-esteem and esteem of others. 'Firmly based', means that the esteem must be based on real capacity and achievements. The need for esteem can be classified in two subsidiary sets. First there is the need for strength, for achievement, for adequacy, for confidence, independence and freedom. These needs are related to self-esteem. Secondly there is the need for reputation, prestige, recognition, attention, importance or appreciation. These needs are about the esteem of others.

A.1.5 Self-actualization

The fifth and according to Maslow highest need is the need for self-actualization. If all four of the lower needs are fulfilled there still is a feeling that something is missing in one's life. It is the tendency of someone to become actualized in what he is potentially. To become everything that one is capable of becoming. A musician must make music, a poet must write, an athlete must sport. Maslow recognized for example Abraham Lincoln, Albert Einstein and Eleanor Roosevelt as self-actualized people. He found that these man were more reality-centered and problem-centered then others. They also had a different perception of means and ends. Ends don't justify means. Means could be ends themselves and the means is often more important than the ends [22].

A.1.6 Eight levels

Later Maslow presented an adapted version of his hierarchy with eight levels [24]. The upper four levels of this pyramid are all based on the former self-actualization level. The new hierarchy never had the impact the five-level variant had. Still the five-level variant is more widespread.

A.1.7 The ultimate human motivation theory

Markus and Kitayama see Maslow's theory as a cultural dependent, not a universal human motivation hierarchy. They found that while American students indeed do associate positive feelings with personal achievements, Japanese students' associate positive feelings with good relations with others [23]. The cultural differences are especially recognized in the higher needs.

A.1.8 Product development and basic needs

As referred to in the introduction of this chapter a product might have a competitive advantage when it helps an individual fulfill a basic need. One way to recognize how a product can do this is to ask this question for each individual need [25]. Of each aspect of the five levels a question will be asked. "How will the new product impact my acquisition and use of food?" is an example from the first level. A negative answer on a question regarding one of the most basic layers has a greater impact then a negative answer on one of the higher levels of needs.

Appendix A

B Former and current mobile devices

In this appendix I will introduce the reader into the history, and current possibilities of mobile devices. The method that is described in this thesis is not only suitable for mobile devices, but as this method is developed in an organization where mobile applications are being developed, the method will for sure be used for the development of context aware mobile applications. This appendix will give some insight in the world of mobile applications for those who do not have any experience with mobile applications but are curious for the possibilities. First I will shortly describe the history of mobile devices. After that I will talk about different aspects of a mobile device and their current state.

B.1 History

In 1946 the first mobile phone was used [9]. This was the first mobile communication device that was connected to the telephone network. The Swedish police was able to make six calls before the car battery ran out. It took twenty-five years before the first truly successful public commercial mobile phone network was launched in 1971, called ARP. Nowadays this network is sometimes called the zeroth generation (0G) network. In 1981 the first generation mobile network 1G was launched. It was first launched in Saudi Arabia, years later in Finland. It was a set of analogue mobile phone standards. In 1984 the first truly mobile phone became available by Motorolahttp://www.abc.net.au/widebay/stories/s1428825.htm
[10]. It cost $ 3.995,- but was such a revolution that there was a waiting list of several thousands. It featured one hour of speaking time and eight hours of stand-by time and weighed two pounds.

Figure 0-2 – Motorola DynaTac 8000X. The world's first truly mobile phone

The second generation of mobile phones (2G) was introduced in 1990 in the United States. The main improvement was the switch from analog to digital voice communication. GSM is the most popular second generation standard for mobile phones. The Global System for Mobile communication is used by over 2 billion people in more than 200 countries [11]. GSM is an open standard, which means that although the standard is widely used for years already, improvements in the standard are still being made for new networks and devices, which are backwards compatible to the older devices. An example of a further development of the GSM standard is the introduction of GPRS, described below.

Although voice was the main component, the first release had three basic components: Voice, Fax and Supplementary Services [12]. One of those supplementary services was the Short Message Service. It almost didn't make it into the GSM standard [13] but with 500 billion messages a year in the mid-2004 [14] it can be called a great success.

In the 1997 release of the GSM standard, GPRS was introduced. The General Packet Radio Service made it possible to use a GSM mobile phone to browse the Internet [15]. This version is also called 2.5G because it bridges the gap between the voice based second generation mobile phones and the data based third generation mobile phones. The services associated with 3G provide the ability to transfer both voice data (a

telephone call) and non-voice data (such as downloading information, exchanging email, and instant messaging) [16]. 3G should have been one standard, but there turned out to be three different camps. UMTS was developed for the current GSM users, CDMA2000 for the American, Korean and Japanese countries not using GSM and TD-SCMA in the Republic of China.

B.2 Current mobile devices

B.2.1 Network

3G is currently being rolled out and already available in the dense populated areas in almost all the western countries. But according to Forrester, the market share of 3G users on the mobile phone market in Europe will still only be 10% in 2007 [17]. While 3G is being rolled out, also 2.75G is introduced. 2.75G is also known as EDGE. EDGE is an upgrade of the GPRS standard [18]. The main improvement is the higher maximum data rate of up to 473 kbps instead of the 160 kbps of GPRS. EDGE can be made available a lot faster because it uses the GSM network, where 3G needs an expensive new network. Therefore EDGE may be deployed as a low-budget alternative for 3G, or as a replacement for 3G in areas where the population is too small in order to deploy 3G without losing a

B.2.2 Connectivity

One of the demands for a device to be a mobile device in my definition, is that is connectable to a mobile phone network. Nowadays mobile devices often are able to connect to other entities in other ways.

B.2.3 Infrared

The Nokia 5140 series for example can connect to a pc or a Polar Heart rate monitor sports watch using Infrared [26]. The devices often conform to standards published by the Infrared Data Association [27]. Infrared makes it possible to send data over short distances wireless. It is necessary to have the sensors at short range without obstacles in each others line of sight with an eye to eye angle of thirty degrees at most [28].

B.2.4 Bluetooth

The last few years Bluetooth gained popularity in the mobile phone world. Bluetooth also connects devices wireless over a low distance. The advantage of Bluetooth above infrared is that Bluetooth uses radio technology instead of an optical technology. This makes it possible to connect devices over a low distance no matter whether they’re in

each others line of sight or not. A Bluetooth device can detect surrounding Bluetooth devices and make a connection with them.

B.2.5 Wireless Lan

The market for Wireless Lans has grown from 0.3 billion US dollars in 1998 to 1.6 billion dollars in 2005 [29]. Its increasing popularity has some effect on mobile devices. Although the technique is pretty energy consuming, there are some mobile devices in which WLAN functionality is integrated.

B.2.6 USB

The Universal Serial Bus connects several devices with a wire to a host. The host often is a personal computer. A mobile device can be one of the devices connected to this host. For a lot of mobile devices special cables are available to connect the mobile device to a computer. This interface is often used to synchronize an agenda or to edit logos and ring tones.

B.2.7 Processing power

In personal computers processor speeds are increasing more and more. The same counts for mobile devices. The more advanced mobile devices of the moment have the processing power of a mid-90's PC [30]. In that time it became possible to play music and movie files, browse the internet and play 3d games. Now these applications also become available on mobile devices. But the current mobile devices use only 1 percent of the energy a mid-90's pc did.

B.2.8 Operating system

With the growing processing power a handset can do more then just facilitating a phone call. The operating systems on these mobile devices are getting more and more advanced. Currently there are two major operating system vendors. The first, Microsoft, has a lot of experience with operating systems on personal computers [45]. The second, Symbian, is born in 1998 as a joint venture of Nokia, Sony Ericsson, Motorola and Psion, four of the largest handheld manufacturers in the world [46]. The current are making versions of both systems make it possible to manage files, use hardware that is build into the device like Bluetooth and install a lot of different programs. A modern mobile device can be compared with a personal computer within a handheld.

B.2.9 More then just calling

A mobile device has the special property that it's almost always near the owner. This makes it very suitable as a basis for an "all-in-one-"device. A lot of functionalities can be and therefore are being included in the mobile device. The Nokia 7710 referred to

earlier, includes Personal Digital Application functionalities like a task-list, contacts manager, calendar, alarm clock and Excel, PowerPoint- and Word-support. It also includes messaging applications, like the Short Message Service (included in almost every cell phone), Multimedia Messaging Service and Email. It includes a digital camera with a resolution that can be compared with the size of a modern computer screen, an fm-radio and a multimedia player which support all kinds of audio (MP3, AAC, Real Audio, …), images (JPG, GIF, BMP, …) and movies (MPEG4, Real, …). Modern mobile devices like the 7710 even support Internet Browsing. But of course, it can still be used to make a call.

B.2.10 Memory

Some of these functions need some additional space for files to be placed. A media player for example needs media files. Mobile devices with these functionalities often feature some internal memory, 90 Mb for the Nokia 7710, and a memory extension slot also used in digital cameras. The 7710 can handle MMC memory cards with a maximum size of 1 Gb.

B.2.11 User Interface

More advanced mobile device features like a digital agenda, video player and internet browsing ask for larger screens. Email, a digital agenda and Word processing ask for a large keyboard. On the other hand, portability also is important which keeps the devices small. Not only do larger screens take more space, they also are more expensive and vulnerable. Nokia mainly uses four different screen sizes, each with other possibilities.

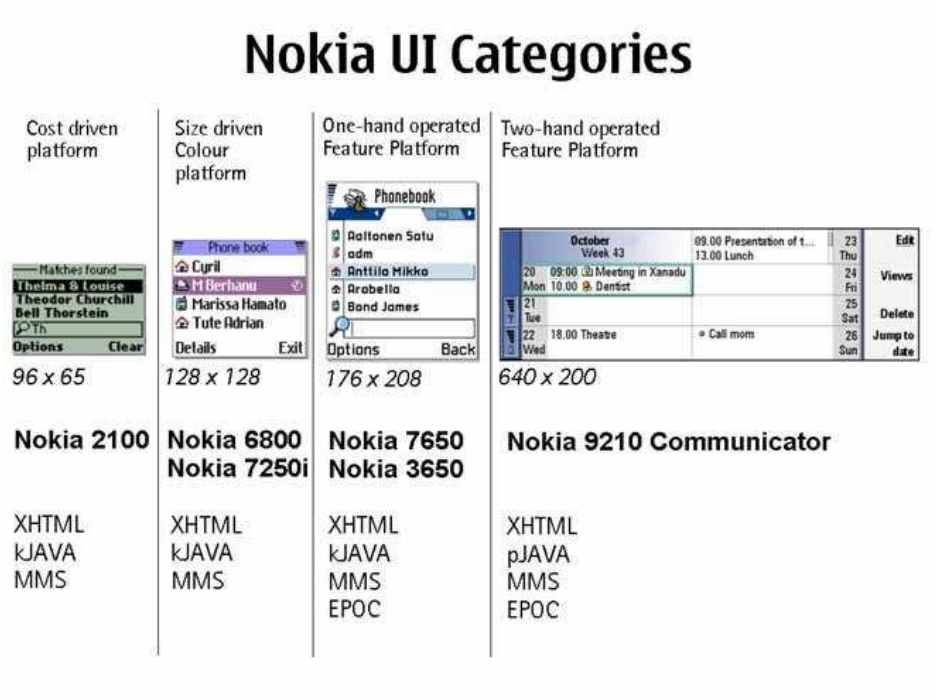


Figure 0-3 – Nokia screen sizes

Device manufacturers struggle to get the screen size, keyboard size and mobile device size optimized. Nokia left a keyboard away on its 7710. A stylus pen combined with a touch screen has to make a keyboard obsolete. Sony Ericsson combined a large screen and a stylus with a keyboard that can be twisted above a part of the screen on the p900. Motorola folds the screen and keyboard of the Razr V3 together to keep the device small. The 7710 has a screen size of 640 * 380 pixels, the P900 has 208 * 320 pixels and the Razr V3 has 176 * 220 pixels in its main screen.

Figure 0-4 – Nokia 7710 Figure 0-5 - SonyEricsson P900

Figure 0-6 – Motorola Razr V3

B.2.12 Battery power

Where the Motorola Dynatec had a standby time of 8 hours, current devices do a lot better. The Nokia 3510 for example has a standby time of thirteen days while it only weighs 105 grams [31]. Battery power still is under development though. Where mobile devices get larger displays, more features and greater processing power, battery consumption grows. The Nokia 7710 for example needs to be reloaded twice a day in extensive use [32].

B.2.13 Device size

As referred to before, the size of a mobile device has a lot to do with the size of the screen and the keyboard. That is illustrated with the smallest device of Nokia until now. The Nokia 6100 is offered as a very good and reliable business phone with a color screen. Although it has a standby time of 320 hours, it only weighs 76 grams and with 102 x 44 x 13,5 mm it has a volume of 60 cc [33].

B.2.1 Mobile application structure

Mobile applications often have a client-server structure. There is a difference between a thin client application and a thick client application [3].

B.2.14 Thin client applications

Thin client applications are often called browser based applications. Internet sites can be viewed using a browser on your mobile device. The content can vary enormously without having the need to program new software for the mobile client. An example of a thin client application is the Sony Ericsson portal for mobile phones [5] through the wireless application protocol [6]. A user can browse to this portal on the Internet with his Sony Ericsson mobile phone. He will find specific functionalities made available by Sony Ericsson for a Sony Ericsson mobile device user. The same user can also visit the Yahoo! portal for mobile users [7]. On this website most of the Internet website functionalities are made available for mobile users. This way, an enormous amount of services can be accessed with only a browser present on the mobile device.

B.2.15 Thick client applications

Thick client applications are not dependent on their internet connection because they work offline. Most of the functionality of the program is present in the ‘thick’ client. Therefore they are more reliable. Examples of thick client applications are build in games on a mobile device, like the famous Snake on Nokia phones. But also a music player, a Word document reader or an offline browser are thick client applications [8].

B.2.16 Peer to peer

A third group of server-client structure for mobile devices is the peer-to-peer (p2p) structure. In this structure the only function for the server is to connect two clients. This is primarily used for the main purpose of most of the mobile devices at the moment, phoning. Also SMS (short message service) can be seen as a peer to peer service as a message is exchanged from one client to the other, but this service could also be seen as a thin client application as a message is sent to the server where it is put in a buffer until it can be forwarded to the intended receiver.

C Categorization of context information in ‘user’ and ‘application’

Both categorizations in this appendix will illustrate Dey’s explicit naming of the contexts of the user and the context of the application to be relevant to the user’s task.

C.1 Xerox’s categorization of context information

The people at Xerox used three categories [37]:

- user context (user profile, location, people nearby, social situation, activity, health conditions, agenda setting, etc.);
- execution context (network traffic, status of the device, availability of resources, communication costs, nearby resources, etc.); and
- environment context (weather, light, noise level, temperature, time, etc.).

C.2 Freeband’s categorization of context information

The people at Freeband use a more detailed categorization in their Awareness project [38]. They have a focus on technology instead of on the user like Xerox has. This can be seen in Figure 0-7.

- Access Network context
  properties of (wireless) networks in the neighborhood of the user’s device, such as availability, domain, cost and bandwidth.
- Device capabilities
  capabilities of available devices, such as display size, memory, computational speed, communication interfaces, available multimedia encoders/decoders and battery power.
- Service infrastructure context
  includes user-subscription details, security information, user identity, access right policies, privacy aspects for managing and exchanging (parts of the) stored context.
- Application context
  characteristics of the applications currently available or used at the user’s device and the applications the user is currently subscribed to.
- Session context

information necessary to maintain running sessions while roaming, including description of sessions currently in use by the user, what kind of data is communicated and resource requirements.

- User context

  describes the (remaining) person's situation, environment, preferences and history, include spatiotemporal information (location, time, task, role, activity), physiological context, mental state (anxiety, happiness, excitement) and social domain (persons nearby).

C.3 Comparing Freeband's and Xerox's categorizations

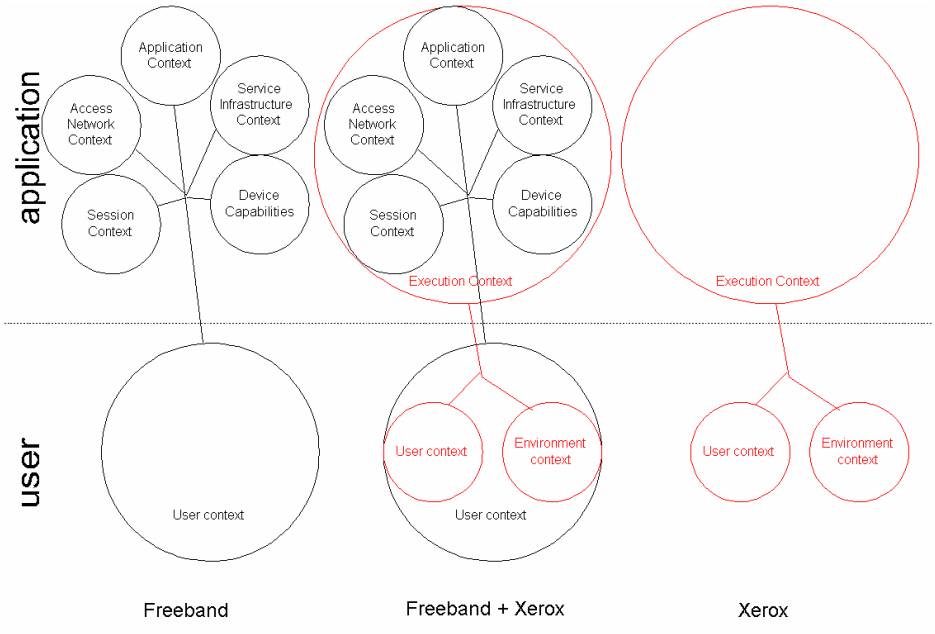


Figure 0-7 – Comparison Freeband and Xerox categorization

The categorizations of Xerox and Freeband overlap. The overlapped categories form two large groups of context information. These are the groups that Dey also presented in his definition of context; application and user. Both can be related to the user's task and both should be related to the user's task in order to find relevant context information.

D Current context aware mobile applications on the market

Only a few years ago a mobile phone was just a handheld telephone. The possibilities to measure or retrieve context information and use it in an application are growing. The following examples are just an indication of what is currently on the market. It is not a complete list of all the context aware mobile applications. As an illustration to what context aware mobile applications are on the market currently, I will give some examples. More information on the technologies named here can be found in appendix A.

D.1 Low level

One of the first context aware mobile applications on the market was that a user could change the way he gets noticed about an incoming text message or phone call. The noise can for example be switched off and vibration can be turned on when in a meeting. This is a low level of context awareness because the user must manually enter the context information, but according to the definition of Dey, it is context awareness. The definition of context awareness was:

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

The described functionality uses the context information that the user is in a meeting to turn off the sound and turn on the vibration; this is a relevant service because the user does not want to be disturbed by the sound of his mobile phone.

D.2 Personalization

A user can configure his device according to his own demands. This way the device will use the context information "preferences of the current user" for the ringing tone, screensaver, color of the display, language of the interface and many other things. This form of context awareness is low-level because the user should manually configure the device to his needs. The above example of turning of the sound when in a meeting is an example of personalization. The user preferences are stored in the profile "meeting" and he selects this profile when he is in a meeting.

D.3 LBS

A higher level of context awareness can be found in a ‘location based service’ (LBS). The American Global Positioning System (GPS) [53] can be used to determine the position of a person or device. It is used by navigation system, which are more and more running on mobile phones in combination with a Bluetooth GPS receiver [54] [55]. Current mobile devices with a mobile internet connection can use traffic information to guide a user around a traffic jam or suggest a restaurant if the user searches a place to eat.

Another technology for location determination is the use of the mobile phone network. With the signal strength from a mobile device to surrounding mobile phone towers, the location of a mobile device can be determined. The accuracy is by far not as good as GPS, but it gives a good indication where to look when for instance your mobile phone equipped child is gone [56]. It also works inside building, where GPS needs a free view to the sky.

A lot of context aware mobile applications at the market are location based services. This is probably due to a number of reasons. At first, a location is relatively easy to measure using one of the systems mentioned above. Secondly, it is easy to make good use of the sensor-information. If the location of a device is X, this information can be used as input for an application. No assumptions have to be made. Thirdly, there is almost no user input required in location based services. If for example a navigation system is used, only the destination has to be entered and during the entire voyage, context information, like current location, is used. The problem with for example status information about a user is that the user often has to update its status himself. An example is changing the status of an instant messenger to “out to lunch”.

D.4 Networking

Once, there was only one kind of mobile phone network through which a user could make a call. Now, there is the standard GSM connection for calling, but a mobile phone could also connect to the mobile internet with GPRS, EDGE or 3G. It can sometimes use connections like Bluetooth, Wireless Lan, or Infrared to connect to other device or through these other devices to the Internet. A mobile device that supports for example 3G, recognizes the availability of a 3G network to connect with, when the users has a membership on a 3G at his telecom provider. The device uses the context information

of available networks and notices the user of these available networks. This doesn't only account for 3G, but also for Bluetooth and Wireless Lan for instance.

Nokia developed an application using Bluetooth to find other people within 30 meters of the user who also have this application installed on their Bluetooth Nokia phone [57]. Through this application people can meet each other and exchange files, messages and information.

Appendix D

E The sources of context information

An application can use different context information and the same context information can be used by different applications. In order to simplify the tasks of creating, maintaining and extending context aware systems, the University of Helsinky introduced a context infrastructure with multiple layers [42][Figure 0-8]. Application and context information are separated and a service layer is put between them. This service layer makes it possible for an application builder to use context information without worrying how to get it. If it is available, the service infrastructure can deliver it. The service infrastructure collects context information from different context providers and makes sure that the applications on top of it can use this information. A context provider can have its own sensor which measures context data or it can get its data from another context provider. One special way of getting context data is that the context provider is build into the application. The usage of an application by a user gives information about the user. Usage of these inference patterns are described by Kleinbauer [74].

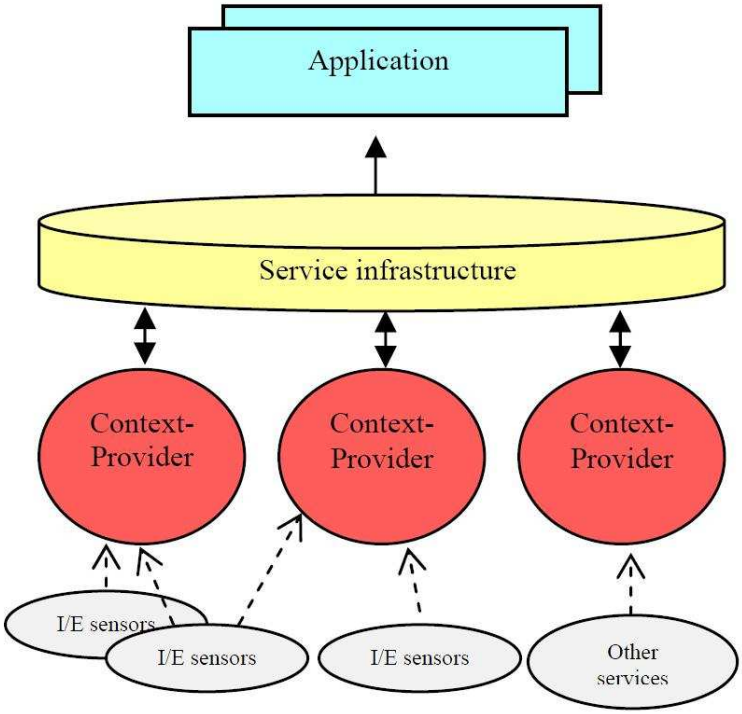


Figure 0-8 – Context infrastructure by the University of Helsinky

F Descriptions of the considered task analyzing techniques

In this attachment I will give descriptions of the eight task analysis techniques considered for using in the method for finding context aware functional requirements. The criteria used for selecting a useful task analysis technique are mentioned in paragraph 6.4.1. In the sequencing paragraphs, the chosen technique is described in more detail.

F.1 Charting and network techniques

Graphic descriptions of tasks can be made by charting and network techniques. There are numerous types of charting and networking techniques. These techniques help create a very clear representation of a system of tasks, but as the complexity of tasks increase, the size and complexity of the graph will also rise enormously.

F.2 Decomposition Methods

Decomposition methods start with a global description of activities. This description is used to determine what other information needs to be gathered with the decomposition methods. These methods are very well suited as a basis to for example an information requirements list. A disadvantage is that the methods are very time consuming, not only in itself, but a first analysis has to be done already before the start with a decomposition method.

F.3 Link Analysis

Link analysis is used to study communication links between objects, whether these objects are individuals or an application. This is particular interesting in situations where transitions play an important role. The technique is not hard to use and it is objective due to its observable and measurable data.

F.4 Timeline Analysis

Kirwan and Ainsworth [64] define timeline analysis as ‘an analytical technique for the derivation of human performance requirements which attends to both the functional and temporal loading for any given combination of tasks’. The functional requirements cover what a human operator needs to do to complete a task. The temporal requirements

describe how quickly it has to be done. The technique is easy to use in building as well as interpreting an analysis due to its graphical representation.

F.5 Operational Sequence diagrams

A sequence of control movements and/or information collecting activities, which are executed in order to accomplish a task, is called an operational sequence. Such an operational sequence can be represented in a diagram. These diagrams are very useful for showing the relationship between operations.

F.6 Hierarchical task analysis (HTA)

A hierarchical task analysis represents tasks in a hierarchical structure of tasks and goals, with scenario dependent sequences of tasks to fulfill the goals, presented textually or diagrammatically. This method starts with establishing the purpose of the task analysis and from there, deciding the rules where the analysis should stop. HTA is economical with time, because it is easy to learn and lets the analyst only develop the parts of the hierarchy where it is justified. The hierarchical structure is easy to comprehend and gives a quick understanding of the represented system of tasks. This makes it suitable in an early stage of application development, where the functionalities of the system have to be made up. The HTA method works top-down, which guarantees a thorough analysis downwards.

F.7 Hierarchical task analysis + middle out

The Husat Research Institute presents a way to work out the hierarchical structure of tasks in HTA, starting with a random task in the task structure [68]. Instead of working top-down, they work from the middle out. This has the disadvantage that the analysis might become less thorough as the direction where the analyst is heading is still unknown. But on the other side, the advantage is that it is possible to search for higher goals in a task analysis.

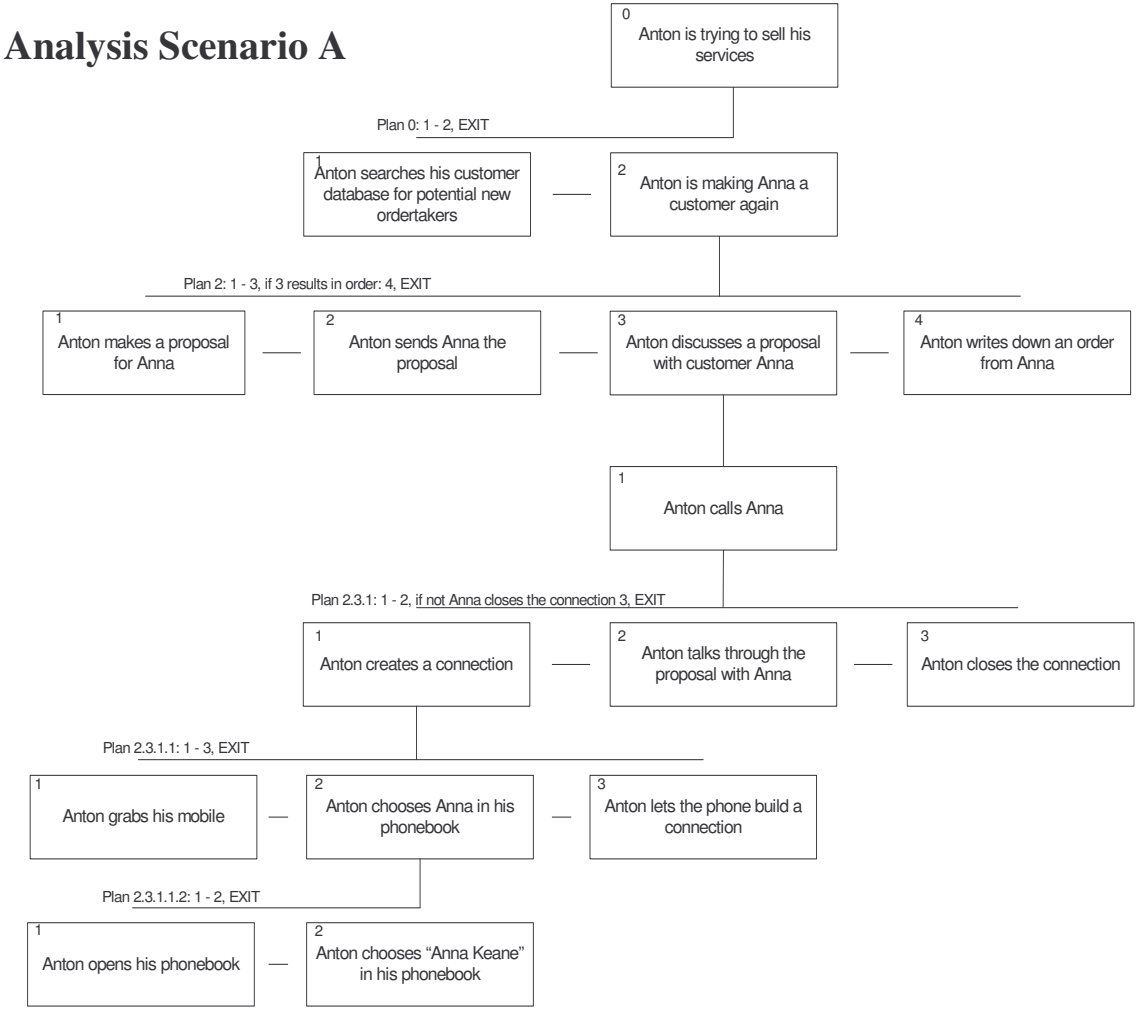
F.8 GOMS

Goals, Operators, Methods, Selection rules (GOMS), is like HTA a method which explicitly supports hierarchies of tasks [78] and works top-down. GOMS is not a single method, but it is a family of methods where ‘goals’ are reached in terms of ‘operators’. These operators can be seen as the earlier described subtasks. Operators are elementary perceptual, motor and cognitive acts. Methods are the sequence of tasks to reach a goal and selection rules are similar to the support of different scenarios in HTA. The most

important difference with HTA is, that it has more rules to follow, which results in a better structured task analysis, especially useful in detailed analysis of for example usage of an existing application. It takes more knowledge and time of the analyst however for building as well as using the task analysis.

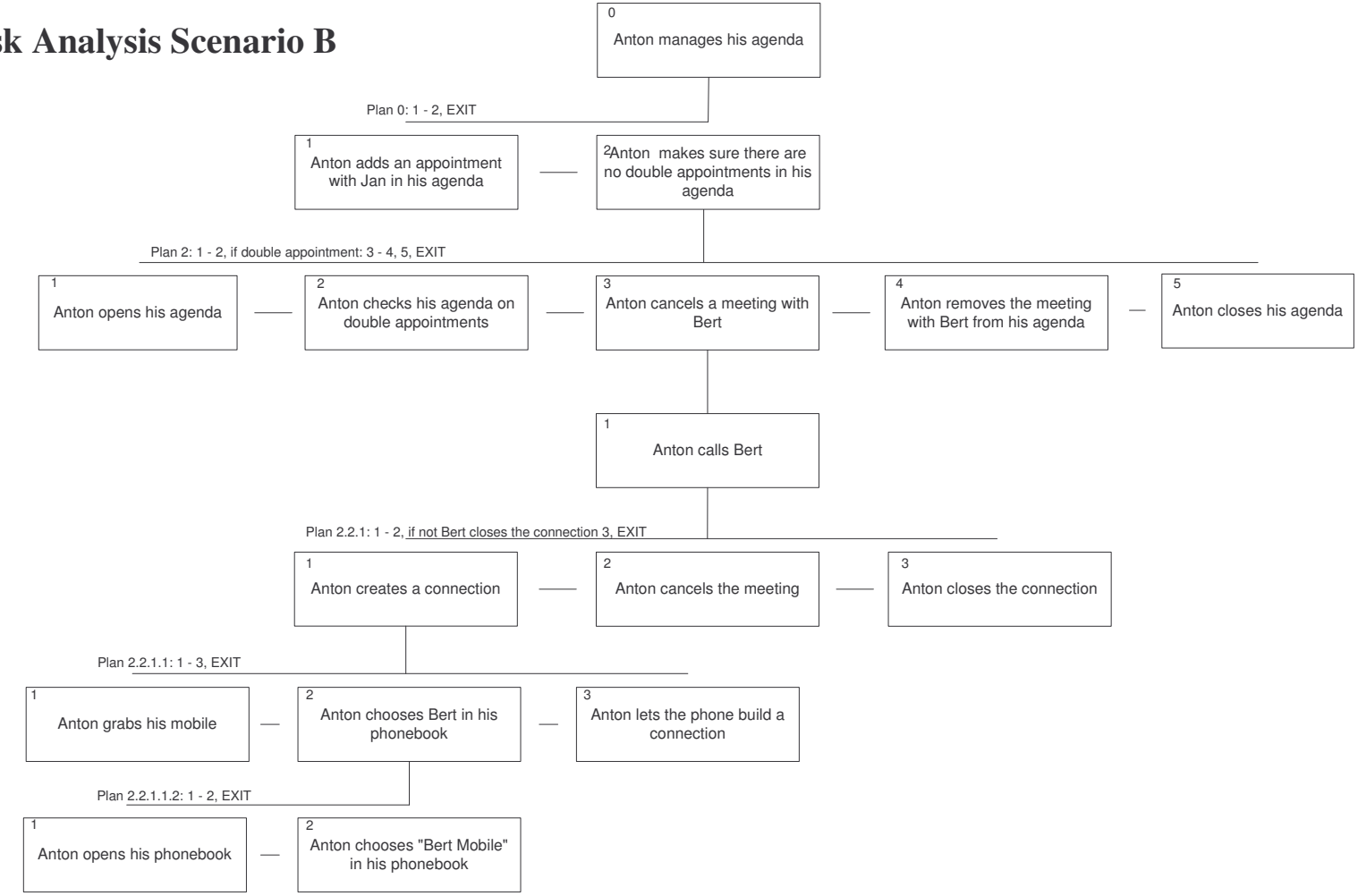
Appendix G

G Task Analysis Scenario A



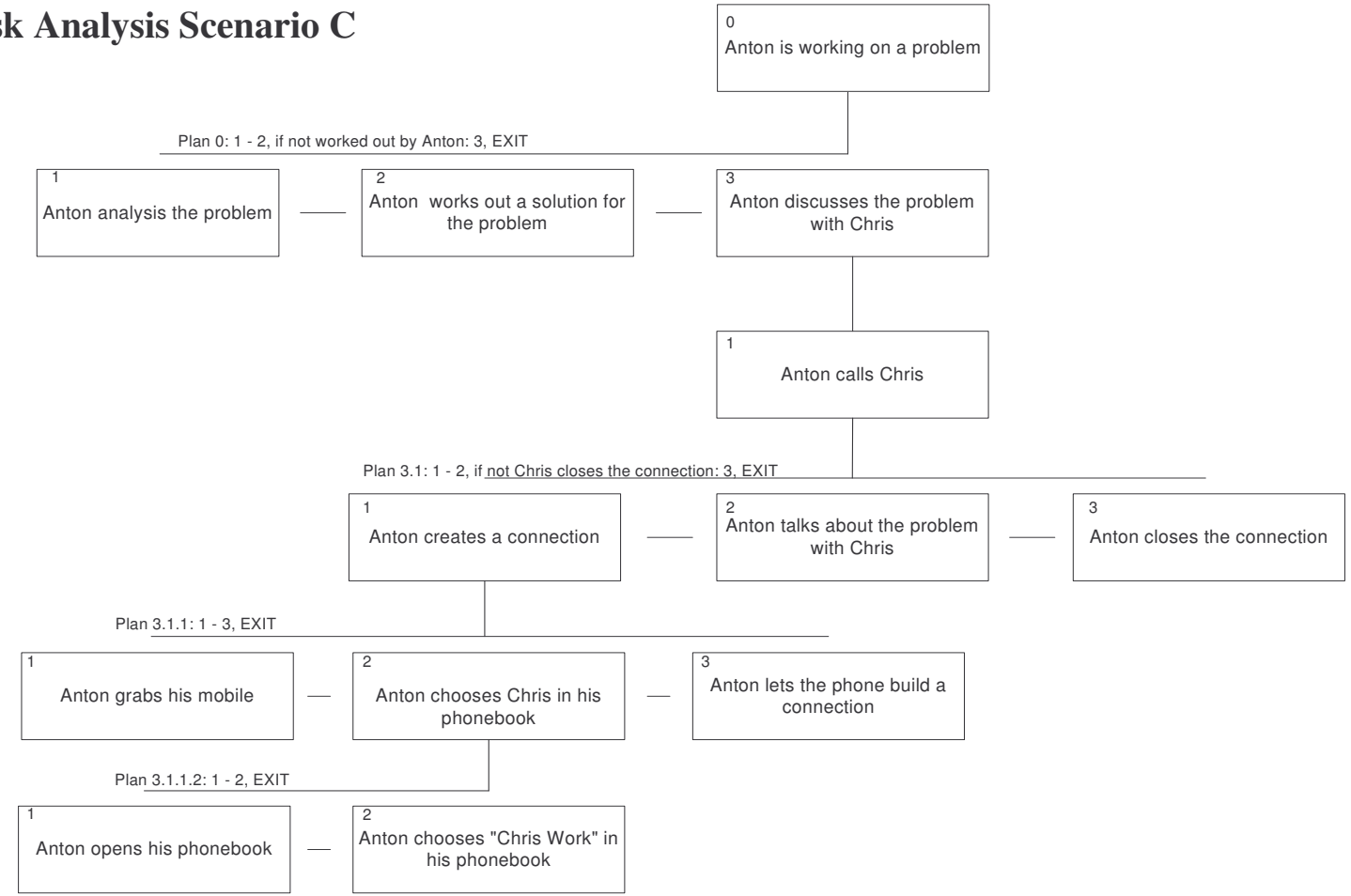
Appendix G

H Task Analysis Scenario B



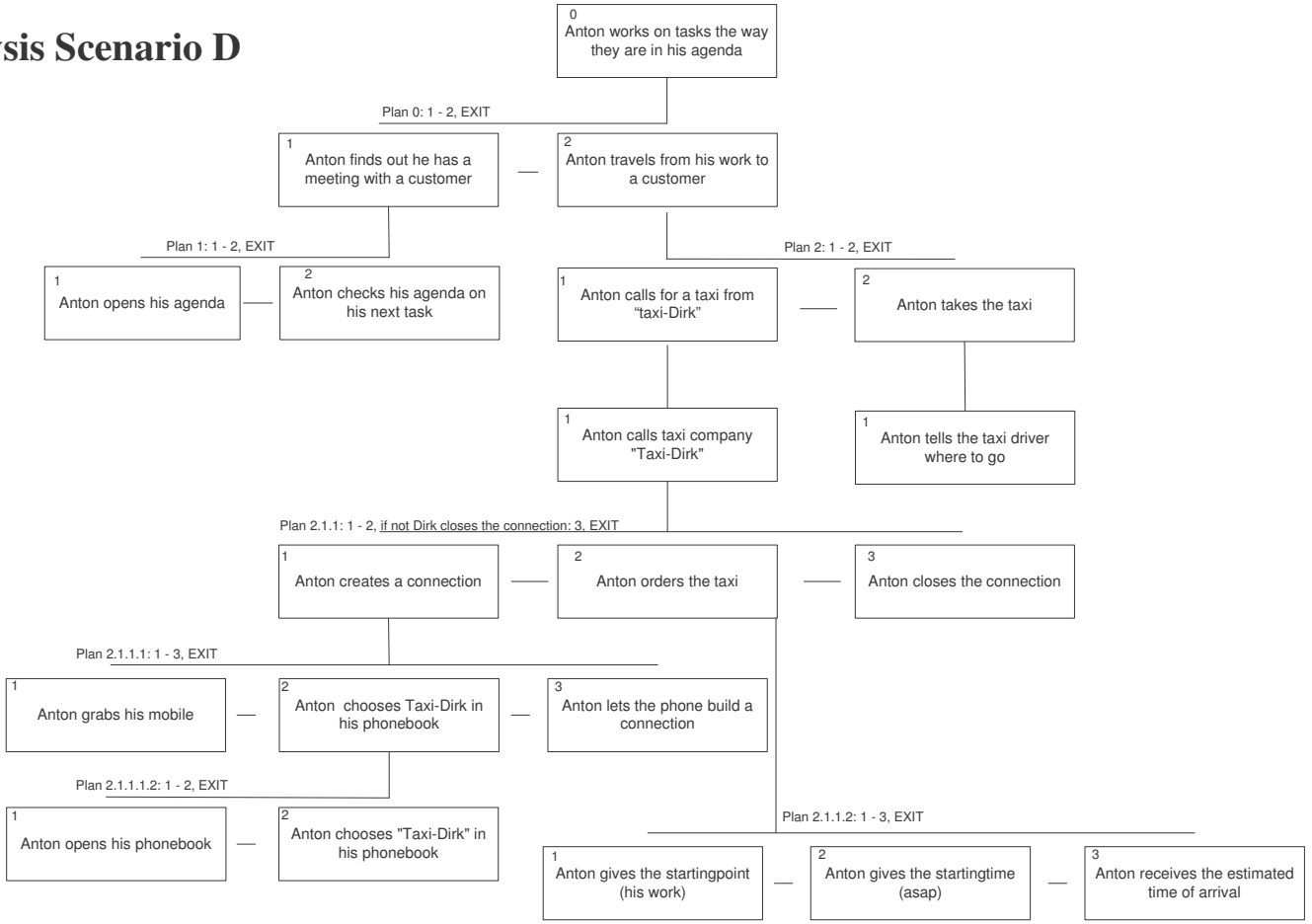
Appendix H

I Task Analysis Scenario C



Appendix I

J Task Analysis Scenario D



Appendix J

K Functional decomposition of available devices

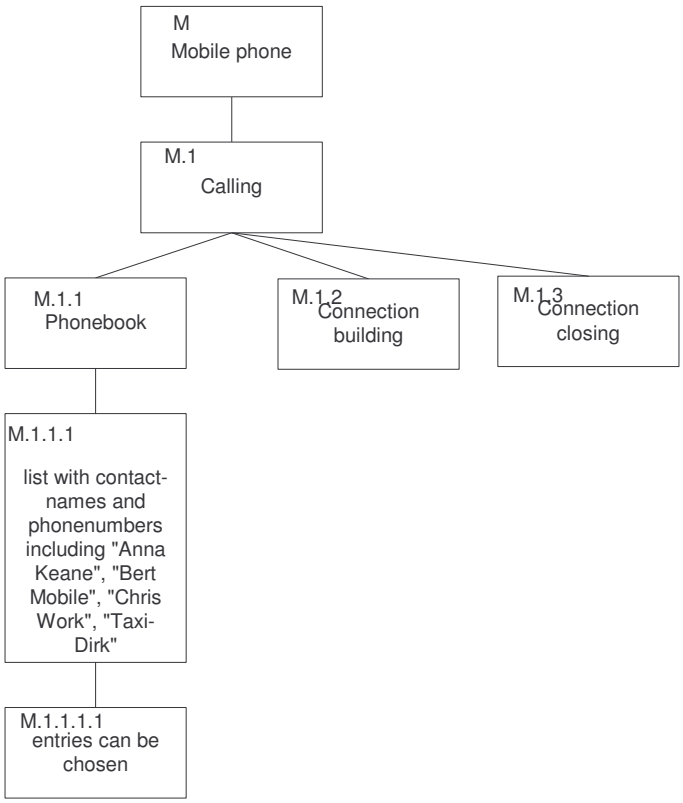


Figure 0-9 Mobile phone

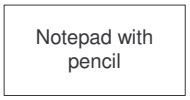


Figure 0-10 Notepad with pencil

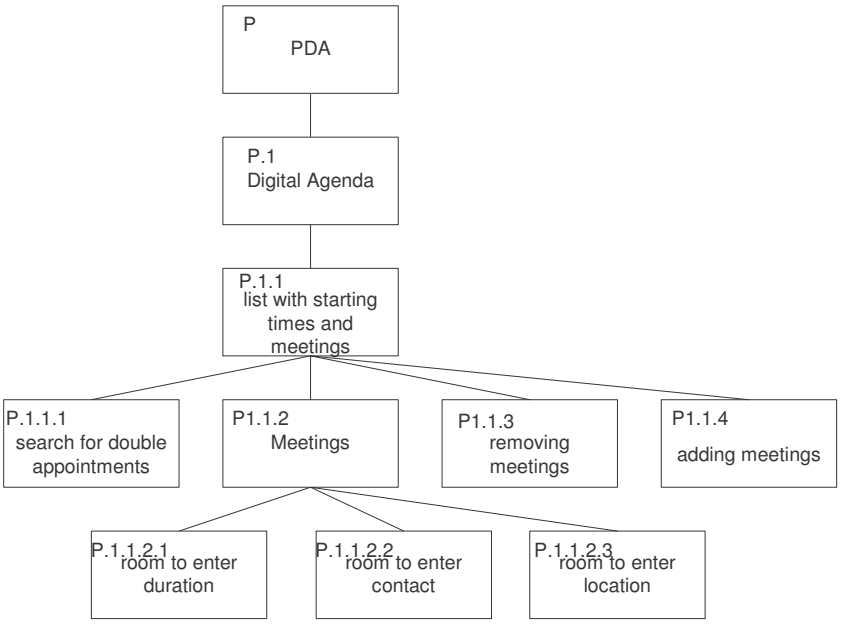


Figure 0-11 PDA

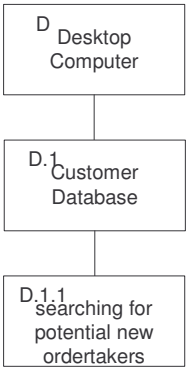


Figure 0-12 Desktop Computer

L Functional requirements for scenario A

L.1 Information and services presented to the user according to the user's context

I analyzed the information-need for all decisions that are made in the task analysis.

A useful application would be to give the information needed to support the making the decision at the moment that the decision is to be made. I did not add the secondary goals found here to the task analysis, because I did not want to add complexity to the task analysis.

Task Decision Functional Requirements

1 Anton searches his **customer database** for new potential order takers.

-What people are potential customers for the services?

No context aware functional requirements

-What other applications have contact information of potential customers?

A1 While using an application with contacts, a shortcut to other applications with contacts could be presented in the user interface of the device used. The context information used is that the application is aware of the need of the user to search for contacts and presents information according to that.

2 Anton is making **Anna** a customer again

-Who have the highest potential to become a client again?

No context aware functional requirements

-Who might become the most beneficial customers?

No context aware functional requirements

-What potential customers can be reached the easiest?

A2 When using a list of contacts, the availability of each contact could be displayed. The context information here is the availability of the contacts.

-What is the mood of the potential new customers?

A3 When using a list of contacts, the mood of each contact could be displayed. The context information here is the mood of the contacts.

-Is the function of the person in the company still the same?

A4 The current function of a person in an organization could be made available. This current function is context information.

2.3.1 Anton **calls** Anna

-How can Anna be reached?

A5 The ways how someone can be reached at a moment could be made available to a user. This is context information about that someone.

-Is Anna at work at the moment?

A6 The current location or of Anna could be made available to the user. The location of Anna is context information about her. The agenda of Anna could be made available to an authorized user. This agenda represents context information about Anna.

-How does Anna prefer to be reached?

A7 Anna's preferred communication channel could be set by Anna and made available to her contacts. The context information is how she wants to be reached.

-How does Anna prefer to discuss proposals?

A8 Anton could have noted the best way to contact Anna for a proposal. This information could be displayed by the device if it senses or if Anton notes that he wants to discuss such a proposal.

2.3.1.1.1 Anton decides to grab his **mobile**

-Do I know her phone number out of my head?

No context aware functional requirements

-Which phone devices can be used?

A9 If Anton makes clear on his mobile device that he wants to contact Anna, different devices to contact her with could be presented. This is context information about what phone devices are available for both Anton and Anna and which of these can contact each other.

-What are the costs for using these devices?

A10 The cost for using the different devices to call Anna could be presented. The context information used is the different devices to be used, the most likely devices to be called and the network and corresponding costs used for it.

2.3.1.1.2.2 Anton chooses **"Anna Keane"** in his phonebook.

-On what phone can Anna be reached?

A11 The different phones that Anna can be reached on could be displayed next to her name in the phonebook.

A12 The preferred phone to be contacted on could be set by Anna, distributed to her contacts and become the default phone number to call Anna on at that moment. The context information is on what phone Anna prefers to be contacted on.

-What are the costs of calling Anna on her difference phones?

A13 The cost for calling Anna on the different devices could be presented. The context information is the target phone of Anna and the network and corresponding costs used for calling her there.

Then I analyzed how information needs as a task itself can be supported.

In task 1 Anton needs a list of potential customers in order to contact them. He wants to have this list available when he asks for it. The potential of contacts to become customers again will depend on whether the contacts have a need for Anton's services, whether they are in a good mood to give an order, whether they have money for the service, whether they are at work or not, and so on.

A14 It will be useful for Anton to know what the mood of his contacts is and whether they are at work or not to determine whether there is a change that they will buy a product again. This information can be presented with his list of contacts or with his customer database.

The final step is to recognize the needed services in the task analysis and analyze in what context the services are used.

The services used in this task analysis are the service to present a list of potential customers; in this case a customer database, and the mobile phone. The phone has a phonebook.

The phone is used to call someone whose name was on the list with customers.

A15 It would be useful to have the ability to call Anna directly from this list. This can be possible by combining the phone, phonebook and customer database in one device, where the contacts are objects who you can call from anywhere in the application.

A16 An alternative of integrating phone and customer database is that they share each others contacts and are aware of the tasks being applied in the services. The customer database could for instance let the phone know that if Anton is going to make a call, the chances are good that he will call Anna. The phone could make 'calling Anna' a suggested option in its interface.

L.2 Automatic execution of a service when in a certain context

One should wonder what tasks could be automated and what context information is needed for that.

Task Automation Functional requirements

1 Anton searches his customer database for potential new order takers

Anton now searches through the customer database for potential new customers. This could be automated it the database knew the criteria that Anton uses. This is not a context aware service. The information need for automating this task is not known at this moment. Therefore no context aware functional requirements can be defined.

2 Anton is making Anna a customer again

A17 An application could be aware of the services need of Anna. If Anna is in need of a service Anton delivers, a standard proposal, adapted to Anna by the preferences as stated by Anton, will be send to her which she would only have to confirm.

2.1 – 2.2 Anton makes a proposal to Anna | Anton sends Anna the proposal

A18 A standard proposal is adapted to Anna by the preferences as stated by Bert.

2.4 Anton writes down an order from Anna

A19 The proposal could be send to Anna in a way that she would only have to fill in the quantity of services and confirm.

A20 The mobile phone could listen to the conversation and notice that Anna confirms the proposal and with what conditions. Anton could review what was concluded and confirm with one push on the button.

2.3.1.3 Anton closes the connection

A21 The connection could automatically be closed if there was no talking anymore for some time.

2.3.1.1.2 Anton chooses Anna in the phonebook

A22 As Anton selected Anna to be called as a renewed customer, his phone could have suggested calling Anna on a prominent place on the interface of his mobile phone.

L.3 Tagging of context to information for later retrieval

One should look where information is retrieved in a task analysis and whether it would have been useful that it was tagged.

A23 Information is retrieved in task 2.3.1.1.2. Anton there retrieves the phone number of Anna. It might be useful to tag this phone number with context information. The number is however already tagged with the name of Anna and the type of phone the number belongs to. Other context information can be tagged and

might be useful in some scenarios, but there are no reasons within this scenario to add other context information to the number.

M Functional requirements for scenarios B

M.1 Information and services presented to the user according to the user's context

I analyzed the information-need for all decisions that are made in the task analysis.

A useful application would be to give the information needed to support the making the decision at the moment that the decision is to be made. I did not add the secondary goals found here to the task analysis, because I did not want to add complexity to the task analysis.

Task	Decision	Functional requirements
2.3	Anton decides to cancel a meeting with **Bert**.	
	-What is the importance of both (double) meetings?	
		No context aware functional requirement
	-How important are the meetings?	
		B1 It could be made possible to add a priority rating to the meetings. Anton could then view this rating for all his meetings to decide which is the most important..
	-How urgent is the meeting?	
		B2 Meetings could be grouped into projects. The date of the next meeting or deadline in the project could be used as a measure for the urgency of the meeting. This date could be presented along with the meeting.
	-How convenient is it to both other persons that the meeting is cancelled?	
		B3 It could be made possible to view the priority rating as stated by the other persons who have this meeting. This could be used to decide how important the meeting is.
	-How important are they both to me?	
		B4 The names of the other participants could be made available to the meeting.
		B5 A rating could be added by Anton to his contacts on importance.

-How easily can I contact them to cancel the meeting?

B6 The currently available ways to contact the participants to the meetings could be added to the participants' names viewed with the meeting.

2.3 Anton decides to **cancel** a meeting with Bert.

-What is the importance of the meeting?

B7 It could be made possible to add a priority rating to the meetings Anton could then view this rating for all his meetings.

B8 It could be made possible to view the priority rating as stated by the other persons who have this meeting. This could be used to decide how important the meeting is.

-How urgent is the meeting?

B9 Meetings could be grouped into projects. The date of the next meeting or deadline in the project could be used as a measure for the urgency of the meeting. This date could be presented along with the meeting.

2.3.1 Anton decides to **call** Bert

-How can Bert be reached?

B10 The ways how someone can be reached at a moment could be made available to a user. This is context information about that someone.

-How does Bert prefer to be reached?

B11 Bert's preferred communication channel could be set by Bert and made available to his contacts. The context information is how she wants to be reached.

-Can't I remove the meeting from his agenda myself?

B12 It could be possible to let Anton remove the meeting from Bert's agenda. Bert should in that case authorize Anton to have writeable access to Bert's agenda. Bert's agenda could propose to Anton to remove the meeting from both their agenda's.

2.3.1.1.1 Anton decides to grab his **mobile**

-Do I know his phone number out of my head?

No context aware functional requirements

-Which phone devices can be used?

B13 If Anton makes clear on his mobile device that he wants to contact Anna, different devices to contact him with could be presented. This is context information about what phone devices are available for both Anton and Bert and which of these can contact each other.

-What are the costs for using these devices?

B14 The cost for using the different devices to call Bert could be presented. The context information used is the different devices to be used, the most likely devices to be called and the network and corresponding costs used for it.

2.3.1.1.2.2 Anton chooses **"Bert Mobile"** in his phonebook.

-On what phone can Bert be reached?

B15 The different phones that Bert can be reached on could be displayed next to his name in the phonebook.

B16 The preferred phone to be contacted on could be set by Bert, distributed to her contacts and become the default phone number to call Bert on at that moment. The context information is on what phone Bert prefers to be contacted on.

-What are the costs of calling Anna on her difference phones?

B17 The cost for calling Bert on the different devices could be presented. The context information is the target phone of Bert and the network and corresponding costs used for calling him there.

Then I analyzed how information needs as a task itself can be supported.

In this example there is no information retrieval task.

The final step is to recognize the needed services in the task analysis and analyze in what context the services are used.

The services used in this task analysis are the (digital) agenda and the mobile phone. The phone has a phonebook.

B18 The phone is used to call someone whose name was in the agenda. It would be useful to have the ability to call Bert directly from the agenda. This can be possible by combining the phone, phonebook and agenda in one device, where the contact are

objects who you can have a meeting with, or call from anywhere in the application.

B19 An alternative of integrating phone and agenda is that they share each others contacts and are aware of the tasks being applied in the services. The agenda could for instance let the phone know that if Anton is going to make a call, the chances are good that he will call Bert. The phone could make 'calling Bert' a suggested option in its interface.

M.2 Automatic execution of a service when in a certain context

One should wonder what tasks could be automated and what context information is needed for that.

Task Automation Functional requirements

1 Anton adds an appointment with Jan in his agenda

B20 The agenda could be set opened for trusted persons as defined by Anton, to create meetings with Anton themselves. Context information here is the trusted persons of Anton and the ability to view the other person's agenda.

2 Anton makes sure there are no double appointments in his agenda

B21 The agenda could check for double appointments itself and cancel meetings according to their priority as defined by Anton. The meetings could be cancelled in the agendas of the other persons if authorized or an email / text message could be send. For meetings sooner than a time as defined by Anton, or with people as defined by Anton, a list of suggested adaptations is presented to Anton with shortcuts to proposed actions. Context information is here the double appointment, priority of meetings, authorization of access to other ones agenda, possible ways to contact another person, access to other ones agenda itself and the current task of Anton where information is represented according to.

2.2 Anton checks his agenda on double appointments

B22 The agenda could check for double appointments itself. A list of suggested adaptations, depending on priorities set by Anton, is presented to Anton with shortcuts to proposed actions. The context information here is the double appointments, the priorities set by Anton and the current task of Anton.

2.3 Anton cancels a meeting with Bert

B23 The meeting could be cancelled in the agendas of the other persons if authorized or an email / text message could be send. For meetings sooner then a time as defined by Anton, or with people as defined by Anton, a list of suggested adaptations is presented to Anton with shortcuts to proposed actions. Context information is here the authorization of access to another person's agenda, the other person's agenda itself, the time and people as defined by Anton and the current task of Anton according to which the list is represented.

2.4 Anton removes the meeting with Bert from his agenda

B24 The agenda could use voice recognition to determine whether the appointment is canceled over the phone and cancel it if it is. Context information is here whether the appointment is indeed cancelled or not.

2.3.1.3 Anton closes the connection

B25 The connection could automatically be closed if there was no talking anymore for some time. Context information is here whether there still is a conversation or not.

2.3.1.1.2 Anton chooses Bert in his phonebook

B26 As Anton was busy in his agenda with the meeting with Bert, his phone could have suggested calling Bert on a prominent place on the interface of his mobile phone. Context information is here Anton's previous task (he was busy with the meeting with Bert).

M.3 Tagging of context to information for later retrieval

One should look where information is retrieved in a task analysis and whether it would have been useful that it was tagged.

B27 Information is retrieved in task 2.3.1.1.2. Anton there retrieves the phone number of Bert. It might be useful to tag this phone number with context information. The number is however already tagged with the name of Anna and the type of phone the number belongs to. Other context information can be tagged and might be useful in some scenarios, but there are no reasons within this scenario to add other context information to the number.

N Functional requirements for scenario C

N.1 Information and services presented to the user according to the user's context

I analyzed the information-need for all decisions that are made in the task analysis.

A useful application would be to give the information needed to support making the decision at the moment that the decision is to be made. I did not add the secondary goals found here to the task analysis, because I did not want to add complexity to the task analysis.

Task	Decision
3	Anton decides to **discusses** the problem with Chris -Is the problem too complex to solve alone?
3	Anton decides to discusses the problem with **Chris** -Which contacts have knowledge on the subject of the problem? -Which contacts have time to discuss the problem? -Which contacts are willing to discuss the problem with Anton? -Which contacts are available to be contacted at the moment? -Which contacts are part of the organization that the problem is relevant to?
2.3.1	Anton decides to **call** Chris -How can Chris be reached? -How does Chris prefer to be reached?
2.3.1.1.1	Anton decides to grab his **mobile** -Do I know his phone number out of my head? -Which phone devices can be used? -What are the costs for using these devices?
2.3.1.1.2.2	Anton chooses **"Chris Work"** in his phonebook. -On what phone can Chris be reached? -On what phone does Chris want to be reached?

-What are the costs of calling Chris on his difference phones?

Then I analyzed how information needs as a task itself can be supported.

Anton needs the solution for a problem. There are multiple ways to find solutions for problems. There is to little information available to give alternatives to support finding a solution for Anton's problem.

The final step is to recognize the needed services in the task analysis and analyze in what context the services are used.

The services used in this task analysis include a mobile phone. The phone has a phonebook and is used to call someone. In the scenario there are no other services used and therefore no services are in the context.

N.2 Automatic execution of a service when in a certain context

One should wonder what tasks could be automated and what context information is needed for that.

Task	Automation	Functional requirement
3.1.3	Anton closes the connection	*The connection could automatically be closed if there was no talking anymore for some time.*

N.3 Tagging of context to information for later retrieval

One should look where information is retrieved in a task analysis and whether it would have been useful that it was tagged.

Information is retrieved in task 3.1.1.2. Anton there retrieves the phone number of Chris. It might be useful to tag this phone number with context information. The number is however already tagged with the name of Anna and the type of phone the number belongs to. Other context information can be tagged and might be useful in some scenarios, but there are no reasons within this scenario to add other context information to the number.

It might be useful to save the conversation with Chris as the solution on Anton's problem might be in it. It will then be useful to

tag this conversation with context information about the date and time of the conversation. In order to find the information back, all the context information around the conversation can be useful, because Anton could search for the conversation with things he remembers about the context the conversation was in.

O Functional requirements for scenario D

O.1 Information and services presented to the user according to the user's context

I analyzed the information-need for all decisions that are made in the task analysis.

A useful application would be to give the information needed to support the making the decision at the moment that the decision is to be made. I did not add the secondary goals found here to the task analysis, because I did not want to add complexity to the task analysis.

Task	Decision
2.1	Anton decides to order a **Taxi** -What alternatives for traveling towards the customer are there? -How far is the customer away? -Are there public transport facilities available in the neighborhood of the customer? -Are there public transport facilities available in the neighborhood of Anton? -How late does Anton have to be at the customer? -Does Anton have to be able to work on his way? -What are corporate rules for such transportations? -How much time does it take to get there for each alternative? -How much does it cost to get at the customer using each of the alternatives?
2.1.1	Anton decides to call **Taxi Dirk** -What are the costs using different taxi companies for this ride? -What is the current availability of the competitive companies' taxis? -What is Anton's preferred taxi company? -Does Anton's work have a standard taxi company?
2.1.1	Anton decides to **call** Taxi Dirk -How can Taxi Dirk be reached? -How does Taxi Dirk prefer to be reached?

-What is the fastest way to order a taxi at Taxi Dirk?

-What is the most reliable way to order a taxi at Taxi Dirk?

-Can't Anton put a reservation straight into the system of Taxi Dirk?

2.1.1.1.1 Anton decides to grab his **mobile**

-Do I know the phone number out of my head?

-Which phone devices can be used?

-What are the costs for using these devices?

Then I analyzed how information needs as a task itself can be supported.

In this scenario Anton needs to give his work as a starting address (task 1.1.2.1). He might not know the address of his work out of his head. It would be useful to have a functionality that gives the address corresponding to the current geographic location of the user and his mobile device.

Further on, Anton needs to tell the taxi driver where to go (task 2.2.1). This can be supported by providing the address information belonging to the customer with whom the meeting was planned on his PDA. On entering the taxi, this information could be provided on the PDA screen.

The final step is to recognize the needed services in the task analysis and analyze in what context the services are used.

The services used in this task analysis include a mobile phone. The phone has a phonebook and is used to call someone. In the scenario there are no other services used and therefore no services are in the context.

The services used in this task analysis are the PDA and the mobile phone. The phone has a phonebook. The phone is used to call someone.

The agenda is used to look for Anton's next task.

O.2 Automatic execution of a service when in a certain context

One should wonder what tasks could be automated and what context information is needed for that.

Task Automation

2.1.1.3 The connection could automatically be closed if there was no talking anymore for some time.

2.1.1.2.1 The starting point of Anton could be made up from the location of the mobile device.

2.2.1 Anton could digitally send the address information of his customer to the taxi driver using his PDA.

O.3 Tagging of context to information for later retrieval

One should look where information is retrieved in a task analysis and whether it would have been useful that it was tagged.

Information is retrieved in task 3.1.1.2. Anton there retrieves the phone number of Taxi Dirk. It might be useful to tag this phone number with context information. The number is however already tagged with the name of Anna and the type of phone the number belongs to. Other context information can be tagged and might be useful in some scenarios, but there are no reasons within this scenario to add other context information to the number.