User Interface Design Based on Object-Oriented Analysis: The Process of Designing a Mobile Communication System

Christian M. Nielsen, Michael Overgaard, Michael B. Pedersen, Sigge Stenild, and Jan Stage

Aalborg University, Department of Computer Science,
Fredrik Bajers Vej 7, DK-9220 Aalborg East, Denmark
{monrad, miov, mbp, stardust, jans}@cs.auc.dk

Abstract: The aim of object-oriented analysis methods is to reveal the essential information from the users' domain of work. This focus is particularly interesting in the design of interactive user interfaces for mobile devices because the severely limited screen space leaves no room for anything else but the most essential information. This paper describes the process of designing the user interface of an interactive location-aware mobile system to support communication in a specific safety-critical application domain. The domain was the fuel department of a coal-based power plant. The process involved ethnographic studies of work activities in the application domain, object-oriented analysis of the problem and application domains, design and implementation of two prototypes, and usability evaluations of each prototype. The description of the process emphasizes the interplay between the object-oriented analysis and the user interface design activities.

1 Introduction

Mobile handheld devices, such as mobile phones and personal digital assistants (PDAs), are spreading fast and are becoming ever more powerful. The mobile technology has provided organizations, as well as individuals, with the ability to work in novel and previously unanticipated ways [25]. A variety of mobile devices are already being used in almost every workspace, to obtain information and interact with other users, especially for communicative purposes [13]. In addition, mobile devices are also used to participate in the social practices of exchange [4, 27]. – [gammel tekst fra det nuværende afsnit 4]

A variety of handheld computing devices such as mobile phones and personal digital assistants are emerging, and many are already on the market. These devices are proving to be a challenge for user interface designers because they provide limited and different means of interaction and small screens.

The key elements of any GUI design project are to determine what modalities are to be used, what the content of each screen should be, and how this content is organized and laid out. In terms of functionality it is important that the required functionality is available in the correct screens and at the right time when the user wants to use the specific function. Other important design decisions are how to position the different elements on the screen and ensure that the elements have optimal attributes like for example size and color. [MBP1]

The literature on user interface design for mobile devices provides numerous examples of specific designs. Typically the rationale behind a design is explained and the design itself is described in detail. However the way in which designers developed the design and the underlying analysis of the domain of use is rarely described in a methodical manner that enables others to learn from the design process [ERCIM 2002]. In this article we briefly describe the rationale of the design, but emphasis is on methodological foundation of the design process.

Methods are generally used with the purpose of solving a problem. A software development method usually consists of a series of activities, being abstract prescriptions on how to reach a state where the problem has been solved. The purpose of a method is to enable practitioners with different backgrounds to avoid repeating mistakes that others have encountered. Thereby, methods enable less experienced people to be more likely in succeeding [6]. Methods cannot be expected to solve all problems, but they can make the life of practitioners somewhat easier.

A key decision in this case study was the selection of the specific method that was used. In this selection, we limited ourselves to object-oriented methods. There are three reasons for this. First, object-oriented methods have become the most widely used approach in the development of interactive software systems [20]. Second, the basic concepts of the object-oriented approach, objects, states, and behaviours, are well suited for describing the system's context [20]. This relation to context is particularly important when we are developing mobile location-aware systems. In addition, object models are central in analysing and understanding the problems relevant to the users. This includes the users' understanding of the system and their envisioned behaviour while using the system [11]. Third, the members of the development team had been using object-oriented methods and programming languages in other projects. Thereby, they had considerable experience with the object-oriented approach.

We have chosen to delimit the experiment to a specific application area: process control and communication in high-risk environments. This area of application was chosen because workers in such environments could often benefit from access to computer-based tools for process control and communication while being mobile. Object-oriented methods for analysis and design of a mobile device used in such a safety-critical domain are also relevant because they can help to identify and focus on essential information. This is imperative because of the limited means of interaction and screen space on the mobile device. An object-oriented method is also well suited since it can explicitly model and describe the objects that need to be monitored and controlled [OOA&D].

In order for the important information from the object-oriented analysis to affect and improve the GUI design process and for the object-oriented approach to become an integrated part of this process it would be desirable to combine GUI design and the object-oriented analysis, thereby creating a combined method. The results of an object-oriented analysis is typically documented through the use of UML therefore the GUI design should also be document using UML.

Hvilke metoder findes; hvordan kan det gøres (forbinde de to ting)

The purpose of this paper is to present how we have designed a handheld mobile device interface supporting communication and process control in a safety-critical domain through the use of parts from different object-oriented methods. Section 2 describes the experimental design process and includes an overview of the methods that were considered as candidates for the experiment. Section 3 presents the user organization where we conducted the experiment. The organization is a coal-based power plant, and the application domain is the work processes in the fuel department. Section 4 provides a description of the method that was used combined with experiences from the application of the method for analysis and design of a mobile handheld device to support communication in the safety-critical domain. [MBP2]In section 5, we discuss general lessons learned through the experiment. Finally, section 6 provides the conclusion.

2 Design Process

The fundamental decision was to employ an object-oriented method for the reasons emphasized above. In addition, our initial intention was to find a single object-oriented method that supported the design process all way through from initial analysis to the creation of the user interface, including the detailed design of screens and other interaction elements.

We identified a collection of articles about different methods that combine object-oriented analysis and design with HCI design [11]. The methods in that collection are listed in the top row of Table 1. In order to select one of these methods, we developed a set of aspects that a method might include. These aspects are listed in the left column of Table 1. All methods in the collection were evaluated in terms of these aspects. The result is shown in Table 1, where a dot in the table indicates that the article that presents the method in question deals with that aspect.

Table 1 illustrates that some methods focus only on a few aspects, while others cover several. Based on this comparison, we chose Wisdom (Whitewater Interactive System Development with Object Models) [24] as the method we would use in our experiment (see the method in column 6). It covers aspects within all of the four main categories. It departs in an object-oriented approach and the presentation of the method includes detailed designs of user interface elements.

The description of the Wisdom method deals with a wide variety of aspects. Yet when we started using the method, it turned out to provide very limited, if any, support to several of the aspects that seemed to be covered. This applied in particular to the object-oriented analysis and the detailed design of the user interface. Therefore, we included fragments of two other methods that provided support to exactly those areas where we experienced shortcomings with Wisdom.

We chose a general object-oriented analysis and design method that was based on the UML notation. The method is denoted as OOA&D [20]. This method was chosen for two reasons. Firstly, it devotes equal amounts of effort to the problem domain and the application domain. Secondly, we had considerable experience with using this method for analysis and design of a broad variety of software systems. OOA&D has a system choice activity where the overall properties of the system considered are decided. After that, there are two separate analysis activities: problem domain analysis and application domain analysis. All of these three activities involve techniques that we employed to support the analysis activities in Wisdom.

Table 1. Overview of methods

	Part I - Participatory design	**Part II** - Scenario - and task-based design			**Part III** - Use-case based design			**Part IV** - User centered design	
	Chapter 1 [McGinnes & Amos, 2001]	**Chapter 2** [Rosson & Carroll, 2001]	**Chapter 3** [Van Harmelen, 2001-2]	**Chapter 4** [Artim, 2001]	**Chapter 5** [Kruchten et al., 2001]	**Chapter 6** [Nunes & Cunha, 2001]	**Chapter 7** [Constantine & Lockwood, 2001]	**Chapter 8** [Gulliksen et al., 2001]	**Chapter 9** [Hudson, 2001]
Project									
Rich pictures	•			•					
Project Size	L	N/A	N/A	M-L	L	S	All	L	N/A
Application domain									
User profiles/Roles			•	•	•	•	•		•
Use-cases				•	•	•	•	•	•
Scenarios		•	•	•			•		•
Stakeholders									•
Problem domain									
Object modelling	•		•	•	•	•			
Examples of object modelling	•		•	•	•	•			
User interface									
Interface modelling	•			•	•	•			
Examples of interface modelling	•					•			
Other									
Based on	DSDM, RAD	OOA	Idiom94	ETP	RUP	UCEP, LUCID, OMT, Spiral Model, Bridge, OVID		RUP, DSDM, ISO 13407	ISO 13407

Wisdom is more explicit about its shortcomings in detailed user interface design. In fact, it refers to another method that is supposed to deal with the detailed user interface design. The method is denoted Bridge [7] and its aim is to support the transition from object models to user interface design. The Bridge method was used to map the interaction spaces that were specified with Wisdom onto a concrete user interface design, which resulted in a non-functional prototype. The non-functional prototype served as an outline for the implementation of a functional prototype. Yet this last implementation is not described in this paper.

Our design process involved three overall activities that we referred to as requirements, analysis and design. These activities and their primary products are illustrated in Figure 1. Below, we provide an overview of these activities. They are described in detail in section 4. The whole process was closely inspired by Wisdom [22, 23, 24]. The Wisdom process is a software process framework based on a user-centered, evolutionary, and rapid-prototyping model specifically adapted for small teams of developers.

The requirements activity focussed on an overall specification of the users' needs, called a system definition. It outlined the key requirements to the system, the main functionality of the system and the conditions for development. The idea with the system was also embodied in a prototype that was evaluated with the prospective users.

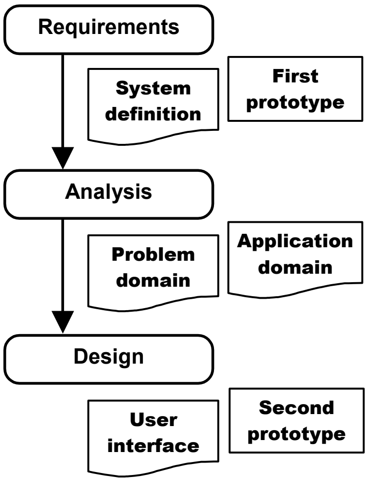


Figure 1. Overview of the design process

The analysis activity focused on descriptions of the application and problem domain. The application domain is where the users are, and the description primarily included user characteristics and use cases. The problem domain is the object of the users' work, and the description of this domain primarily consisted of a general class diagram and a statechart diagram for each individual class.

The design activity focused on the creation of the user interface. This activity produced specifications of various aspects of the user interface. The result was a non-functional prototype that was evaluated with the users. This prototype and the related description of the user interface formed the basis for the implementation of a functional prototype.

3 User Organization and Tasks

A key decision was the choice of user organization. We are generally focussing on safety-critical domains. Therefore, we would prefer a user organization within such a domain. In addition, we stated a list of requirements to the case. Based on these, we chose Nordjyllandsværket, see Figure 2, which is coal-based power plant situated in Northern Jutland in Denmark. The plant produces central heating, electricity, and several by-products that are used in the production of cement. This section provides an overview of the user organization in our development case. This includes a description of the major tasks performed by the employees in the fuel department and the nature of the problems they experience with their collaboration.

We limited the case to the fuel department (see Figure 2) of the power plant. This department is a safety-critical domain where a mobile system will be useful in different locations, and where the employees have different and varying working tasks to solve. Tasks are carried out in collaboration among employees, and during this process the employees must be able to communicate with each other, even if they are not located in the same place.

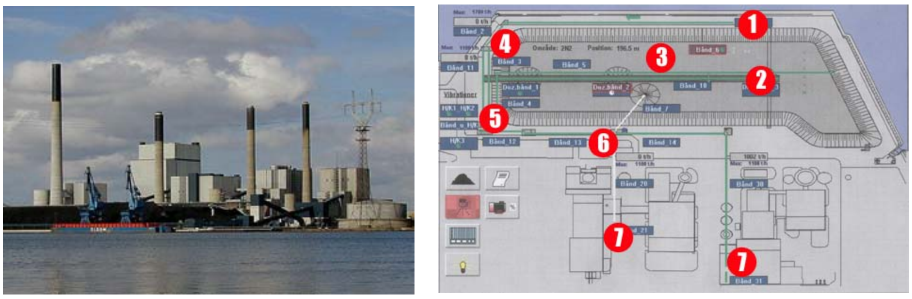


Figure 2. Nordjyllandsværket and the fuel department

The plant is divided into two independent production plants (location #7). The coal to the two plants is supplied from a central storage area (location #2 & #3). The fuel-department is responsible for delivering the coal used in the two production plants, amounting daily to 5000 tons of coal for each.

The employees in the fuel department continuously monitor and control the transportation of coal. They must ensure that the correct amount of coal arrives to the correct location and that the coal has certain properties and quality. To ensure this the coal is processed (location #4 & #5) before transporting it to the final location (location #7). Another important task is that the employees have to prevent the coal from self combusting.

3.1 Communication to Support Coordination

The employees perform a variety of different tasks in order to ensure that the needed amount of coal is delivered to the two production plants. In order to coordinate the many tasks described above the need for quick and easy communication is important, and in some cases even essential in order to carry out the job in a safe and efficient manner. At the present, the devices used for communicating are VHF-radios (walkie-talkies), DECT wireless phones, and some times mobile phones. The control tower (location #6) is the only location where necessary information is accessible and it is also the place where employees can operate and control most of the machinery.

Every element of the coal transport can be controlled through the existing system. When a problem arises, which cannot be solved from the control room, for example in the Grinder building, the person situated here trying to solve a problem do not have access to the information. Furthermore some parts of the machinery can only be controlled from inside the Grinder building. The only way to gain some kind of access to the information systems in the control room is by communicating with a person in the control room either by phone or walkie-talkie.

3.2 Communication Problems

Often the phones are not usable because of the weak signal; hence the only tool for communicating is the walkie-talkie. Several problems are related to the use of this device, since many conveyor-belts run underground, which disrupts the signal, and the machines and conveyor belts are often placed inside concrete buildings, which also disrupts the signal. Finally, there is a deafening noise inside these buildings, which makes talking to each other difficult, and using any kind of mobile devices for verbal communication is virtually impossible.

So basically the employees experienced communication problems related to two areas. One area was explained as problems with connection fallouts when using their communication devices. The second area was problems with noise in some areas of the power plant.

4 Designing the Location-Aware Mobile Communicator

This section describes the process of developing a location-aware mobile communicator to support employeess in the application domain presented above. The description of the process is divided into the three overall activities that were introduced in section 2 and illustrated in Figure 1. The description is based on diaries [15] that were maintained throughout the development process. The diaries were used to retain important information, such as observations, actions, and reflections, and later they served as the basis for reflections on the development process. The development process was carried out during the fall of 2003.

4.1 Requirements

As soon as we began the requirements activity, it turned out that Wisdom provided no significant support to get the process started. The method mentioned what we should do, but no advice on how to do it. To compensate, we added selected techniques from OOA&D that provided the necessary methodological support. The whole activity comprised the following sub-activities:

- Study application domain
- Interiorize project
- Understand system context
- Make user profiles
- Model essential tasks
- Develop a non-functional prototype

- Evaluate the first prototype

The purpose of the first activity was to get an overview of the user organization and their needs for a better communication system. We visited the user organization where we interviewed the employees, saw the work area and tasks. To document this, we took photos of the work places and artefacts, and made video recordings of the way employees carried out key tasks. This gave us an understanding of the application domain and insight into their communication problems, using wireless phones and walkie-talkies in such a noisy environments.

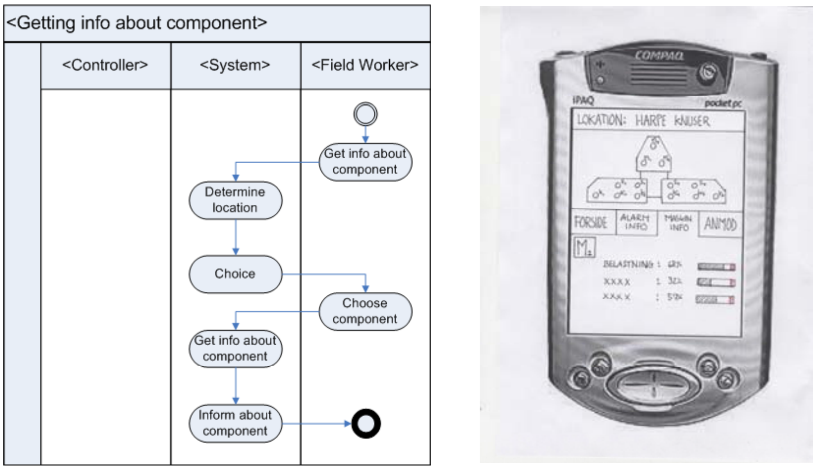
The interiorize project activity dealt with descriptions of the impressions that were gained in the study of the application domain. Wisdom provided no techniques or tools for this activity. Instead, we used two techniques from OOA&D: rich pictures and the system definition. The system definition summarized the overall characteristics of the application domain and the development project in this way:

- Functionality: Communication device with machine state indication and support for communication
- Application Domain: Transport of coal around the power plant, preparation and mixing of coal, monitoring of conveyer belts and problem solving/prevention in production line
- Conditions: Safety critical, noisy environment, dusty conditions, above- and underground, employees have basic IT training/knowledge
- Technology: Pocket PC with Microsoft visual studio 2003 .Net and WLAN
- Objects: Employee, mobile unit, conveyer belt, magnet, screener, grinder, control room computer
- Responsibility: Context-aware mobile communication support system that monitors production line state and facilitates cooperation and communication in a noisy environment

The understand system context activity was conducted for understanding the problem domain of the users. This was expressed in the domain model, an early class diagram that captured the most central objects in the context of the system.

The user profiling was conducted in order to describe the users whose tasks would be supported by the system. We identified two roles: controller and field worker.

Wisdom is an essential use-case and task flow based method. It relies on essential use cases to capture the structure of use patterns and the underlying functional requirements. We had some difficulties making these descriptions, because certain tasks were only conducted rarely, and they did not happen during our visits. To solve this, we staged situations in which we were able to observe the communication between the workers. We identified 9 use cases that should be supported by the system. For each of these use cases, we worked out a diagram that expresses the use case in terms of a statechart diagram. Wisdom denotes this as an essential task flow diagram. Yet in order to preserve consistence with UML, we refer to them as use case diagrams. Figure 3(a) shows an example of a use case diagram.



(a) (b)

Figure 3. The main results from the requirements activity. (a) A use case diagram for getting information about a component such as a machine or a part of a machine. (b) The user interface of the first prototype with a window displaying information about a machine.

To evaluate the outcome of the requirements activity, a non-functional prototype was created (the first prototype). It consisted of a number of screens drawn on paper. Use was simulated by shifting these drawings on the top of the screen of a PDA. An example of the user interface of this prototype is illustrated in Figure 3(b).

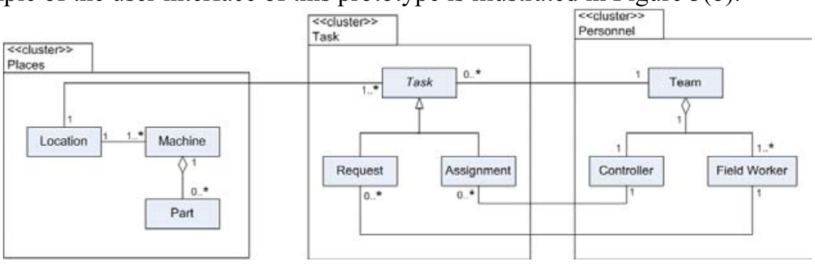


Figure 4. The revised class diagram.

We evaluated this prototype with the users. It enabled us to check whether our understanding of the communication structure was correct. The Wisdom method did not describe which approach to use, when evaluating the non-functional prototype. Based on our experience, we decided to do an informal test at the fuel department, where different employees tried to use the prototype. This led to a discussion about the required functionality and the prototype's structural design. The evaluation resulted in modifications of the use case diagrams, and it yielded useful ideas that should be considered when designing the next prototype.

4.2 Analysis

The analysis activity refined the structure of the system that was described in the requirements activity. One outcome of this activity should be a revised class diagram. The Wisdom method did not provide any methodological support to this. Instead, we used OOA&D techniques to find classes, events, behaviours, and functions, and to produce the required revised class diagram. Thus the analysis activity comprised the following sub-activities:

- Find classes
- Classify events and behaviours
- Find functions

One main outcome of using these techniques was a revised class diagram that depicted the physical locations, the communication, and the users relevant to our system. This diagram is shown in Figure 4. For each class, there was a statechart diagram that described its behaviour.

Function	Complexity	Type
Log in	Simple	Update
Log out	Simple	Update
Get location	Complex	Compute
Check location	Simple	Signal
Monitor location	Medium	Read
Get machine status	Medium	Read
Check machine	Simple	Signal
Get load information	Medium	Read
Check part	Simple	Signal
Get local alarms	Medium	Read
Get global alarms	Medium	Read
Check pre-alarm	Simple	Signal
Check alarm	Simple	Signal
Request stop	Simple	Signal
Request start	Simple	Signal
Request reverse	Simple	Signal
Request assistance	Simple	Signal
Request light	Simple	Signal
Reject assignment	Simple	Signal
Accept assignment	Simple	Signal
Cancel assignment	Simple	Signal
Task done	Simple	Signal

Figure 5. The function list for the system.

The second main outcome was a list of functions that the system should include. This is shown in Figure 5.

4.3 Design

The purpose of the design activity was to create a user interface that fulfilled the requirements specified and detailed in the preceding two activities. In our process, the design activity involved the following sub-activities:

- Interface architecture design
- Internal system design
- User interface design
- Develop a non-functional prototype
- Evaluate the second prototype

Interface architecture design defines the external architecture of the system. This is a key activity as it includes the overall design of the user interface. The main product is a general interaction model that describes a set of interaction spaces and tasks. An interaction space is an abstract user interface elements, and a task is an activity that is carried out by a user by employing one or more interaction spaces.

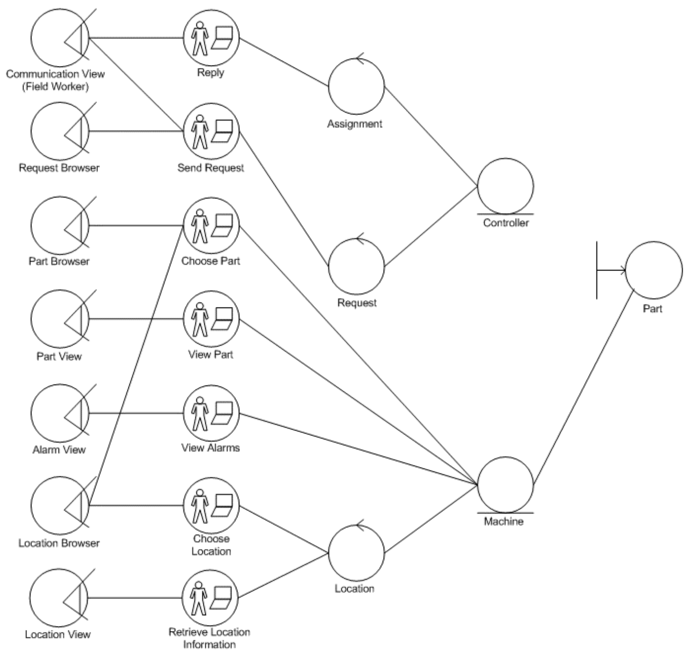
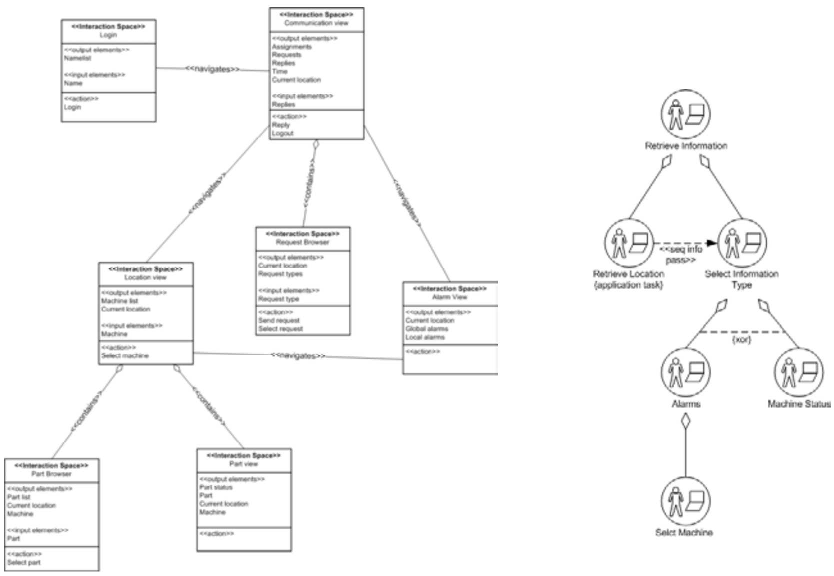


Figure 6. The general interaction model

In this activity we produced an interaction model for each of the essential task flow diagrams from the requirements activity. In doing this, we started out by re-evaluating the use cases and essential task flow diagrams. Each individual interaction model was drawn on the right hand side of the corresponding essential task flow diagram (see Figure 3(a) for an example of an essential task flow diagram). For each activity in the diagram we named the interaction spaces, i.e. the user interface elements, that would be necessary for providing computer support to that activity. In addition, we named the tasks that the user would carry out in that interaction space. When all of the individual interaction models were complete, we combined them into one general interaction model. Figure 6 shows the general interaction model. The left hand column contains the interaction spaces, the next column the tasks and the three columns on the right hand side are internal elements of the system.

Internal system design is mainly a technical task. It focuses on the internal working of the user interface. It involves development of system classes that support the use cases. This relates to the three columns on the right hand side of Figure 6. The main change on the external design of the user interface was that we introduced interactions spaces to log in and out of the system.

User interface design involved the parallel development of two related models: presentation model and dialogue model. The presentation model is a single model that is based on the interaction spaces. The interaction spaces are transformed to classes and they are related to each other with object-oriented structures. The presentation model for the whole system is shown in Figure 7(b). The dialogue model is a collection of individual diagrams that each describe the dialogue when using a part of the system, for example when a user carries out the activities in a task flow diagram. The dialogue model describes the elements and the sequence in the use of the system. One dialogue model is illustrated in Figure 7(b). This is the dialogue mode for the task in Figure 3(a).



(a) (b)

Figure 7. The main results of the user interface design activity. (a) The presentation model for the whole system. (b) A dialogue model the task of getting information about a machine.

The design is based on the use cases that are produced during application domain analysis. The final results are the navigation diagram and window diagrams that are explained in Chapter 8. The purpose of user interface design is to come from this basis to the results.

The first step is to produce a set of interactions models. For each use case, a corresponding interaction model is developed. For each step in the use case, the necessary interaction space and the related task class are defined. The notation used in an interaction model is illustrated on slide 3.27.

The second step is to combine all interaction models into one complete model. This can be done by collecting the elements from the individual interaction models. In this process, there may be repeated task classes and interaction spaces that can be eliminated. An example of a complete interaction model is shown on slide 3.29.

The third step is to describe the elements of the user interface. This is expressed in two types of models:

1. The dialogue models that describe how a single use case is carried out. This is described in terms of the task classes in the interaction model. An example of a dialogue model is shown on slide 4.7.
2. The presentation model describes all the classes in the user interface. The classes come from the interaction spaces. An example of a presentation model is shown on slide 4.9.

The fourth step is to design the individual windows of user interface. A window is a collection of one or more interaction spaces. A method for designing windows is summarized on slide 4.12-4.15. The results are a navigation diagram and a set of window diagrams.

Figure 8. The user interface of the communication device

In the design workflow the shape and architecture of the system are refined for the implementation. The internal system design contains two sub-activities, where the first is to prioritise and select use cases, and the second is to design the use case classes. In the user-interface design activity, the interaction model is transformed into a concrete user interface design. This activity encompasses two concurrent flows of activities, corresponding to the dialogue and presentation components of the interaction model. The user-interface design activity ends by relating both task classes to interaction space classes, hence completing the process of distributing responsibilities between the dialogue and presentation models.

5 Discussion

Husk at vende tilbage til det tema, som titlen på artiklen fremhæver

In doing this, we encountered the fundamental questions of what is the difference between the terms actors and roles and what constitutes a user profile.

When we develop systems for mobile devices, a variety of aspects that are different from those considered in relation to traditional systems, have to be taken into account [18]. Compared to traditional stationary desktop systems, mobile systems are unique in several ways: They are often used while moving, they are used in different locations and situations, and they often have very small visual displays, which result in limited interactions styles [8, 9, 16].

These fundamental differences impose new and fundamentally different challenges on the analysis and design of mobile systems. Especially the design of usable user interfaces leads the user interface designer to explore other ways of displaying information to the mobile user. In order to take all of these aspects into account in the development process, it requires a study of human-computer interaction (HCI) processes to support usability of the mobile devices [3]. There is a substantial amount of articles documenting HCI research in relation to mobile systems, but the area of mobile HCI lacks research and development based on a firm methodological foundation, since most of these activities are based on trial and error [17].

Despite any criticism of inconsistence in the models used in Wisdom, they have proven useful in several aspects. They gave a good understanding of the system, when made, and we did end up with graphical user interface design that proved useful in the usability evaluation, a system that managed to support the users' tasks in connection with problem solving. Having said this it should also be noted that the process of moving from the interaction space model to the concrete GUI design seems a bit like pulling a rabbit out of the magician's hat. What Wisdom really helps to do is that through its many models it manages to facilitate the designers in determining what information should be available in the specific windows and what information should be available for user manipulation and a navigational overview. But Wisdom does not help to position, group, estimate whether the cognitive load is acceptable, or what modalities to use. These fundamental challenges in user interface design all need to be taken into account when designing the user interface and are also closely related to the information presented in the window

When transferring the interaction spaces model to a concrete GUI, the Wisdom method refers to the Bridge method, but several things were not taken into account by the latter method. First of all, there was no indication of how the grouping of objects should be done or how general design issues should be implemented. Even more important in our case, it did not take the screen size into account. Again we realized the importance of the first prototype, since it provided the basic ideas for the design.

We discovered that our essential task flow diagrams were too detailed, because it became too complicated to express the related information in the dialogue models. Despite this, we found that the dialogue and presentation models provided an improved overview of the functions and how and which screens were to be implemented.

Prototyping

Wisdom proposes development and evaluation of prototypes between the different workflows in order to evaluate the result. We find this approach to be troublesome, since it contains some problems:

- Elaborating essential use cases and essential task flows, with the aim at avoiding premature design decisions, contradicts the development of prototypes at the end of the requirement workflow, as it forces the development team to make design decisions.

- The lack of descriptions on how to construct a prototype makes the scope and purpose of the prototype unclear.
- The purpose of developing a prototype seems irrelevant, when the method does not specify how the results from the evaluation of the prototype are utilized in the development process.

The transition from the first to the second workflow illustrates these problems. This transition is supposed to involve some kind of non-functional prototype. However, other than suggesting prototype development and test during this workflow, the Wisdom method was not that specific on these issues. First of all, the Wisdom method had focused on *not* making any GUI design decisions until now, and instead focusing on using essential use cases and task flow diagrams, which was a kind of a problem when designing a prototype. Furthermore it did not specify how the results of the prototype test should be utilized in the further development.

The usability of the implemented prototype was tested through field evaluation in cooperation with the employees, whom are the potential users of the mobile system, and through heuristic inspection performed by usability experts. The overall results of these evaluations showed us that the system was indeed usable, and that the employees all said that they would want to use the system, if it were fully implemented.

6 Conclusion

Husk at vende tilbage til det tema, som titlen på artiklen fremhæver

This paper has presented experiences from an experiment where a UML based method was applied to perform object-oriented analysis and at the same time design the user interface. Hence the method applicability in designing a mobile system intended for communication and generating the appropriate GUI was tested. The WISDOM method was supplemented with activities from the OOA&D method, in order to obtain a complete object oriented approach.

Both WISDOM and OOA&D have weaknesses when considered isolated as methods, but in unity they complemented each other. WISDOM supported OOA&D's recommendation of an early prototype, which again provided material for the subsequent modelling of the problem domain. Furthermore OOA&D made vague analysis activities in the WISDOM method more tangible. Opposite WISDOM accentuated which information that had to be included in the GUI and it emphasised the continuous use of prototypes.

A limitation to WISDOM is that it does not specify how initial application domain knowledge is to be obtained. Another shortcoming of the method is that it does not facilitate specific GUI structure and layout design, which in essence is still up to the GUI designers themselves to incorporate into the prototypes. Lastly, the method only provides information on when a prototype is needed, but not on who has to make the prototypes, how they are made, or how changes to the prototype is to be implemented further on in the process. On the overall level, the procedure worked well. The activities were relevant and the results of different activities comprised a coherent design.

The conclusion of this paper is limited by the fact that we have employed a case-based research approach. We have only applied the method once, and we have only developed a system for a specific domain. In addition the WISDOM method utilises a hotel administration, a typical desktop administration system, as an example when explained, where the method in this paper is applied to a mobile communication system. Still, both approaches are successful.

The experiment indicates that the WISDOM method, in combination with OOA&D, is suitable as an OO approach for developing mobile systems, but also that it is not perfect for generating GUI. The described process should be evaluated in different domains, and alternative approaches to the limitations of WISDOM should be assessed.

Acknowledgments

We are grateful to the employees and the head of the fuel department at Nordjyllandsværket, Bruno Andresen, for giving us the opportunity to use the fuel department as the case in our system development process.

References

1. Alrø, H., Kristiansen, M.: Do You See What I Said? Humans Face to Face. (in Danish: Kan du se hvad jeg sagde? Mennesker ansigt til ansigt). Holistic, Roskilde, 1996.
2. John M. Artim (2001). *Entity, Task, and Presenter Classification in User Interface Architecture: An Approach to Organizing HCI Practice.* In Object Modeling and User Interface Design, edited by Mark van Harmelen. Addison-Wesley, 2001.
3. Francesco Belotti, Riccardo Berta, Alessandro De Gloria and Massimiliano Margarone (2003). *Supporting Efficient Design of Mobile HCI.* Mobile HCI 2003, Udine, Italy.
4. Sara Berg, Alex S. Taylor, and Richard Harper (2003). *Mobile Phones for the Next Generation: Device Designs for Teenagers.* CHI2003, Fort Lauderdale, Florida, 2003.
5. Larry L. Constantine and Lucy A. D. Lockwood (2001). *Structure and Style in Use Cases for User Interface Design.* In Object Modeling and User Interface Design, edited by Mark van Harmelen. Addison-Wesley, 2001.
6. Bo Dahlbom and Lars Mathiassen (1993). *Computers in Context – The Philosophy and Practice of Systems Design.* NCC Blackwell, Cambridge, 1993.
7. Tom Dayton, Al McFarland, and Joseph Kramer (1998). *Bridging User Needs To Object Oriented GUI Prototype Via Task Object Design.* In User Interface Design – Bridging the Gap from User Requirements and Design, edited by Larry E. Wood. CRC Press LLC, 1998.
8. Anind K. Dey and Gregory D. Abowd (2001). *Towards a Better Understanding of Context and Context-Awareness.* CHI2000, The Hague, The Netherlands.
9. Mark D. Dunlop and Stephen A. Brewster (2002). *The challenges of mobile devices for human computer interaction.* Personal and Ubiquitous Computing, Volume 6, Issue 4, 2002.
10. Jan Gulliksen, Bengt Göransson, and Magnus Lif (2001). *A User-Centered Approach to Object-Oriented User Interface Design*. In Object Modeling and User Interface Design, edited by Mark van Harmelen. Addison-Wesley, 2001.

11. Mark van Harmelen (ed.) (2001). *Object Modeling and User Interface Design* Addison-Wesley, 2001.
12. Mark van Harmelen (2001). *Designing with Idiom.* In Object Modeling and User Interface Design, edited by Mark van Harmelen Addison-Wesley, 2001.
13. Harper, R. *People versus Information: The Evolution of Mobile Technolog.* Mobile HCI 2003, Udine, Italy.
14. William Hudson (2001). *Toward Unified Models in User-Centered and Object-Oriented Design.* In Object Modeling and User Interface Design, edited by Mark van Harmelen. Addison-Wesley, 2001.
15. Jepsen, L. O., Mathiassen, L. Nielsen, P. A.: Back to thinking mode: Diaries for the management of information systems development projects. Behaviour and Information Technology, Vol. 8, No. 3 (1989).
16. Masao Kakihara and Carsten Sørensen (2002). *Mobility: An Extended Perspective.* Thirty-fifth Hawaii International Conference on System Sciences (HICSS-35, Big Island Hawaii, ed. R. Sprague Jr. IEEE. 2002.
17. Jesper Kjeldskov, Connor Graham, (2003). *A Review of Mobile HCI Research Methods.* Mobile HCI 2003, Udine, Italy.
18. Steinar Kristoffersen and Fredrik Ljungberg (1999). *Mobile Use of IT.* Proceedings of the 19th Information Systems Research Seminar in Scandinavia, 1999.
19. Philippe Kruchten, Stefan Ahlqvist, and Stefan Bylund (2001). *User Interface Design in the Rational Unified Process.* In Object Modeling and User Interface Design, edited by Mark van Harmelen. Addison-Wesley, 2001.
20. Lars Mathiassen, Andreas Munk-Madsen, Peter Axel Nielsen, and Jan Stage (2000). *Object-Oriented Analysis & Design.* Marko Publishing, Aalborg, 2000.
21. Simon McGinnes and Johnny Amos (2001). *Accelerated Business Concept Modeling: Combining User Interface Design with Object Modeling.* In Object Modeling and User Interface Design, edited by Mark van Harmelen. Addison-Wesley, 2001.
22. Nuno Jardim Nunes (2001). *Object Modeling for User-Centered Development and User Interface Design: The Wisdom Approach.* Ph.D. Thesis, Universidade da Madeira, 2001.
23. Nuno Jardim Nunes and João Falcão e Cunha (2000). *Wisdom: A Software Engineering Method for Small Software Development Companies.* IEEE Software, 2001.
24. Nuno Jardim Nunes and João Falcão e Cunha (2001). *Wisdom - Whitewater Interactive System Development with Object Models.* In Object Modeling and User Interface Design, edited by Mark van Harmelen. Addison-Wesley, 2001.
25. Mark Perry, Kenton O'Hara, Abigail Sellen, Barry Brown, and Richard Harper (2001). *Dealing with Mobility: Understanding access anytime, anywhere.* ACM Transactions on Computer-Human Interaction (TOCHI), 2001.
26. Mary Beth Rosson and John M. Carroll (2001). *Scenarios, Objects, and Points of View in User Interface Design.* In Object Modeling and User Interface Design, edited by Mark van Harmelen. Addison-Wesley, 2001.
27. Alex S. Taylor and Richard Harper (2002). *Age-old practices in the 'New World': A study of gift-giving between teenage mobile phone users.* CHI 2002, Minneapolis, Minnesota.