Aalborg Universitet

AALBORG UNIVERSITY

Combining ethnography and object-orientation for mobile interaction design

contextual richness and abstract models

Kjeldskov, Jesper; Stage, Jan

Combining ethnography and object-orientation for mobile interaction design: Contextual richness and abstract models

Jesper Kjeldskov*, Jan Stage

Aalborg University, Department of Computer Science, Selma Lagerlöfs Vej 300, 9220 Alborg East, Denmark

Abstract

There has been a lot of interest in ethnography within human–computer interaction over the last two decades, and its relevance within systems development is today beyond question. However, one of the challenges reported is that ethnography generates findings and knowledge with such contextual richness that it can be hard to transfer into system design. In the light of recent years' push for the use of ethnography within the area of *mobile* human–computer interaction, this challenge has resurfaced and is of renewed importance to the research field. In this article we describe an interdisciplinary combination of ethnography with a structured software engineering method supporting the transition from collected data to design and implementation. We explore this combination through two case studies of mobile system development for supporting distributed work activities within industrial process control. We show that when developing mobile systems ethnographic data is a highly valuable source of input for developing object-oriented models by providing contextual richness, and that in turn, objected-oriented analysis is a highly valuable method for working with ethnographic field data in systems development by supporting the creation of abstract models. Combining the two, we have a method where ethnographic field studies inform core system design.
© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

In a literature review from the early 2000s (Kjeldskov and Graham, 2003), it was found that the majority of research within the area of mobile human–computer interaction was based on applied research and laboratory experiments. One of the consequences of this was that research was poorly grounded in real world activities and subscribed to the assumption that we already knew what problems to solve—just not exactly how to solve them. In response to this, the mobile HCI community was urged to explore the use of more contextually grounded methods, such as ethnography, as illustrated by, for example, the work on mobility in collaboration by Luff and Heath (1998). In a follow-up review from 2010 (Kjeldskov, forthcoming) it was found that although fieldwork is now much more common, there is still not a strong link to design and development. One of the problems with ethnography for informing design and development of computer systems is that it sometimes generates findings and knowledge with such contextual richness that it can be hard to translate into system design.

There has been a lot of interest within human–computer interaction over the last two decades in how ethnography can inform system design (Crabtree et al., 2009), and the challenge of transition from collected field data to requirements analysis has been described many times in the literature for over a decade (Paay et al., 2009; Kjeldskov et al., 2006; Constantine et al., 2003; Viller and Sommerville, 2000; Hughes et al., 1994). With the push towards using ethnography within the area of *mobile* HCI this challenge has resurfaced and is of renewed importance. Hence the aims of our work presented here are similar to that of the references above, but with a particular focus on mobile

*Corresponding author. Tel.: +45 9949 8921; fax: +45 9940 9798.
E-mail addresses: jesper@cs.aau.dk (J. Kjeldskov),
jans@cs.aau.dk (J. Stage).

computing. As our scope for this research we wish to, firstly, address the overall interplay between ethnography and existing object-orientated methods, rather than the development of new specific modelling techniques as in, or example, Viller and Sommerville (1999) and Iqbal et al. (2005). The latter, however, is an apparent opportunity for further research in light of the case studies presented here.

When designing and developing a computer system, a level of formalization is always required at some point in the process. Very often, this is where findings from ethnographic studies are lost or summarized beyond recognition, simply because they do not translate well directly into system specifications. As a consequence, even though ethnographic field studies might teach us a lot about the use context of our future systems, this knowledge is often difficult to trace through to the resulting designs. This is also the case for mobile systems.

In response to this, we have explored an interdisciplinary combination of ethnography with a structured software engineering methodology that is known for effectively supporting actual system development, namely that of *object-oriented analysis and design* (OOA&D) as devised by Mathiassen et al. (2000). The object-oriented analysis and design methodology has proven its value in systems development by producing formalized and abstract models that can be directly implemented in an object-oriented programming language (for an insightful discussion of different methodologies and representations in human–computer systems development including object-orientation see Benyon, 2002). In the specific OOA&D method described by Mathiassen et al. (2000) the development of these models is done as a part of an analysis of a system's *application domain* and its *problem domain*, respectively. Based on concrete experience with this method, it is our position that ethnographic field study data is highly valuable as a source of input for such application and problem domain analyses by providing *contextual richness*. In turn, objected-oriented analysis is highly valuable as a method for working with ethnographic field study data by supporting the creation of *abstract models.* By explicitly combining the two, we have a method where ethnographic field studies can influence the core of a system design. This is what we intend to show in this article.

We present our experiences with the development of two mobile systems using a combination of ethnography and OOA&D. The first case study concerned the operation of a large container ship (Kjeldskov and Stage, 2003). The second one concerned the operation of a large power plant (Kjeldskov et al., 2006). We present the two development process case studies in detail, showing how they each followed the same overall process from ethnographic field studies, through object-oriented analysis and design, to conception of prototype systems that could be evaluated with prospective users. The second case study naturally built on the lessons learned from the first one, and we show how this led to modifications to the method applied.

The paper is structured in the following way. First we introduce related work. We then describe the overall method applied for our two case studies. Sections 4–9 take you through the five phases of our development process for each of the two case studies and describe the two resulting functional prototypes. At the end of each of these sections, we outline what was learned in that phase, and how the method was modified for the second case study based on the experiences with the first one. Section 10 summarizes and discusses the lessons learned for the different phases of the method, and for the application of the method as a whole. Section 11 concludes and outlines further work.

2. Related work

Mobile and wearable computer devices and applications are being developed for a broad variety of use areas, and recent research has devoted much attention to investigating how in particular distributed mobile workers can benefit from mobile computer systems. This includes work settings where people are concerned with computerized information and processes of critical importance remote from their current location. Early examples include distributed process control and error diagnosing in waste-water treatment plants (Nielsen and Søndergaard, 2000), and the use of mobile multimedia for telemedicine and early diagnosing in emergency ambulance services (van den Anker and Lichtveld, 2000). More recent examples include logistic vehicle management at automobile terminals (Rügge et al., 2009), vessel control within the fishing industry (Lumsden et al., 2008), and mobile technologies in hospital settings (Tang and Carpendale, 2008; Skov and Høegh, 2006; Bardram and Hansen, 2004). In order to avoid mistakes and accidents in such safety critical work domains, it is important to understand the state of the systems being operated. This is a classic challenge within human–computer interaction (HCI) and Software Engineering (SE) in response to which Rasmussen (1983, 1986) suggests that computer interfaces should be designed to improve operators' reasoning about the *domain of operation*, and Norman (1990) suggest that systems should support *human interaction* rather than total system automation. These are good, general, recommendations—but in order to achieve this, it is important that system developers have a great deal of rich, detailed, and structured understanding of the use domain, and that the structures, relationships and processes of the use domain are clearly reflected in the core of resulting system designs.

Analysis, design, and prototyping methods are general means for supporting development of user interfaces and software systems, and a lot of attention has been given to their development and refinement for traditional, stationary, computer technologies. For mobile systems, however, although it has been argued that these necessitate re-thinking of established systems development methods (Krogstie et al., 2004), generally very little has been published on the topic. Instead the field of mobile computing is characterized by

research describing specific systems, their technical implementation, their context, and their use (Kjeldskov, forthcoming; Hosbond and Nielsen, 2005; Kjeldskov and Graham, 2003). Exceptions include Sharples et al. (2002) and Mikkonen et al. (2002) who outline a number of differences from the development of traditional software within the area of HCI, and (Hosbond, 2005) who presents a case study of development practice in seven companies within the mobile industry. Given the inherently "situated" characteristics of mobile systems use contexts, an obvious candidate approach for informing technology development is ethnography.

Ethnography in HCI is a methodology originating from the social sciences and used to provide descriptions of human activities in a particular context through studies in their fullest possible context, such as observational field studies and contextual interviews. Although there is currently a trend towards returning ethnography in HCI to its anthropological origins (Crabtree et al., 2009) and focus on the broader study of culture, it is the use of ethnography for informing systems design that we are interested in here. Like Crabtree et al. (2009) we believe that this is done best through structured in-depth analysis rather than through literary accounts based on selective observation and reporting.

In reviewing the literature from the last half a decade it is clear that ethnography, in various forms, has increasingly been applied as a research methodology within the area of mobile human–computer interaction since 2003 (Kjeldskov, Forthcoming). In our opinion this is a positive trend for the research field as it indicates a shift in perspective from a singular focus on technology to a more holistic view on technology-in-use. However, one of the problems with ethnography for informing design and development of computer systems, mobile or not, is that it generates primarily descriptive findings and knowledge with such contextual richness that it can be hard to transfer into specific system design. Looking at recent literature, which reports only very few accounts of any kinds of fieldwork directly informing design and implementation of mobile systems, this still appears to be a challenge (Kjeldskov, forthcoming).

There has been a lot of research into bridging the gap between ethnography and software engineering over the last decade (Paay et al., 2009). The essence of this challenge is that it can be difficult to transfer findings and knowledge from ethnographic studies to system design and implementation (Hughes et al., 1995). This is partly caused by a fundamental difference in thinking and working within the social sciences and engineering. As pointed out by Viller and Sommerville (1999), ethnography deals in "the particular" while software engineering deals in "the abstract". Extending this further ethnographers work with rich and concrete descriptions and understandings, while software engineers work with formal models and abstractions (Paay et al., 2009). In respect to their outputs, "Ethnographers avoid judgements: designers make them" (Paay, 2008).

Hence, HCI researchers have struggled with the challenge of making these two communities of practice function well together in the creation of software systems that are both grounded in contextual richness and abstract models. From a team composition perspective, approaches have been explored that aim to integrate social analyses better into design processes through creation of multi-disciplinary teams comprising both ethnographers and software engineers (Diggins and Tolmie, 2003; Hughes et al., 1995). From a data representation perspective, approaches have been explored that present outcomes from ethnographic field studies in software engineering terms, such as UML notation (Viller and Sommerville, 1999). However, bridging the gap between ethnography and software development remains a current research challenge (Schraefel et al., 2004; Walenstein, 2003; Wiltshire, 2003), as pointed out in the overview of related literature provided by Paay et al. (2009). Motivating further research within this topic, there are several examples where ethnographically based methods have demonstrated usefulness in the requirements gathering phase of software development projects, for example Crabtree (2004), Viller and Sommerville (2000), Simonsen and Kensing (1998), and Hughes et al. (1994).

3. An empirical study of bridging the gap

Between 2001 and 2006 we conducted two related case studies with the aim of developing mobile information and communication systems for supporting distributed work activities within industrial process control. The second case study built on the experiences from the first one in terms of the system developed and the process followed. Both case studies explored the development and use of mobile computer systems devices for making communication persistent and providing formalized representations of a shared, safety-critical, work domain. Both case studies followed the same overall process of combining ethnography object-oriented analysis and design for obtaining contextual richness as well as abstract models as foundations for the design of functional prototype systems. The central activities of the process, and how they influenced each other, are illustrated in Fig. 1.

The development processes involved (1) ethnographic field studies, (2) application domain analysis, (3) problem domain analysis, (4) design and implementation of a functional prototype, and (5) usability evaluation of the system in use. The ethnographic field study activities involved observational studies and contextual interviews (Beyer and Holtzblatt, 1998). The object-oriented analysis and design activities followed the method described by Mathiassen et al. (2000) with additional input from Nunes and Cunha (2001a, 2001b) and Dayton et al. (1998). The usability evaluations followed the methods described by Kjeldskov and Stage (2004) and Kjeldskov and Skov (2007). In the following sections, we describe each of these five steps for each of the two case studies. We describe the

two case studies in parallel, structured by the five steps of the development process in order to show their similarities, how the second study built on the first, and how it was modified (and improved) on the basis of lessons learned.

4. Ethnographic field studies

The starting point for the two development processes was an ethnographic field study of work activities in the specific domain of industrial process control in focus. These are described below.

4.1. Case study 1: the container ship

The first case study emerged from a larger study into work activities in the maritime domain involving computerized process control and information systems on board Maersk-Sealand container ships. Maersk-Sealand operates some of the world's largest container ships of sizes equivalent to 3(1/2) soccer fields (Fig. 2). The operation of such a ship is safety-critical. Especially when manoeuvring inside a harbour, erroneous actions may result in the ship running aground, into the quay, or colliding with other ships. In either case, such collisions would cause serious material damage, potentially severe injuries for personnel, and possible loss of human life.

The field studies were conducted over a period of several months in 2000 and 2001 and involved researchers taking part in a number of voyages on board large Maersk-Sealand

Fig. 1. The five major activities in the two development processes.

container ships and other ships. During these voyages, data consisting of high quality video and audio recordings and written notes regarding work activities, communication and technology use during operation of the ships was gathered through observations and interviews in situ.

Apart from informing new interface design for existing maritime instruments (Andersen and May, 2001), a number of work activities were identified in which the use of mobile computer systems could be useful. These included diagnostic and maintenance work in the engine room, surveying the condition of reefers during voyages, locating personnel in case of accidents, and supporting various distributed collaborative work activities. Of particular interest to us, the field studies identified some general limitations in means of communication and coordination in relation to the operation of "letting go the mooring lines" before departing from harbour. This particular operation was therefore chosen for further investigation, leading to the design and development of the "Maritime Communicator".

4.2. Case study 2: the power plant

The second case study grew out of the first study and an interest in our work on mobile computing from a regional power plant (Fig. 3). The Northern Jutland power plant is a large coal-based power plant situated on the outskirts of Aalborg, Denmark, that produces central heating and electricity to the region. The operation of the power plant is safety-critical because it involves the operation of large industrial machinery and there are particular fire dangers. The collaboration with the Northern Jutland power plant allowed us to further explore some of the design ideas embodied in the Maritime Communicator, and the methodological approach used to develop it, but in a different domain. This led to the design and development of the "Power Plant Communicator".

For the field studies we took the same approach as with the container ships. We wanted to gather rich empirical data about the use domain before engaging in design, which required spending long periods of time in the field—quite literally getting our hands dirty. After initial enquiries, it was suggested by our collaborators at the power plant that we focus on work activities in the fuel department, as the operation of this part of the plant is essential for ensuring continuous production of energy. As the fuel department also involves workers distributed over a large physical area,
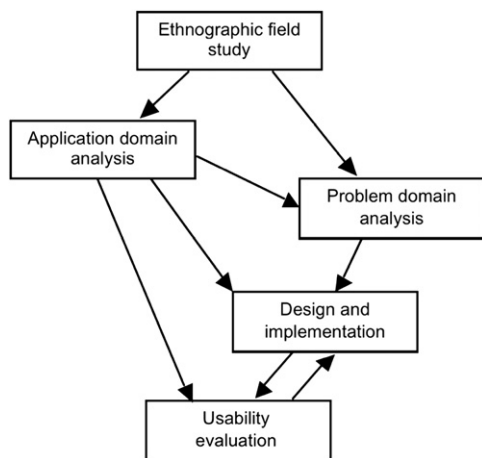

Fig. 2. Sine Maersk at the terminal in Gothenburg. Numbers indicate positions of cooperating teams.

Fig. 3. The Northern Jutland power plant.

relying on centralized computerized controls and coordination based on spoken communication in a noisy environment, this was comparable to the operation of "letting go the lines" on board the container ships.

The field studies consisted of a series of visits to the fuel department of the power plant interviewing the workers in situ and observing their work areas and tasks. This took place over a period of two months. The visits were documented with photographs of work places and artefacts, and through video recordings of the way key tasks were carried out. This provided a rich understanding of the application domain and detailed insight into the communication problems experienced during the work.

5. Application domain analysis

One of the key characteristics of the OOA&D method devised by Mathiassen et al. (2000) is the distinction between *problem* and *application* domain in the analysis phase. Mathiassen et al. (2000) defines the problem domain as "the part of a context that is administrated, monitored, or controlled by a system". Accordingly the application domain is defined as "an organization that administrates, monitors, or controls a problem domain". This distinction is introduced in order to guide the focus of the analysis activity towards the broader use context (application domain) and the specific part of this context that the system is going to be concerned with (problem domain), respectively. The sequence of the application and problem domain analyses is not prescribed but should be informed by the specific system development case and process at hand. When building on top of an ethnographic study, starting with the application domain analysis facilitates a gradual narrowing of focus from the broader context to the specific system, and was found most suitable.

The central principles of the application domain analysis are to describe the current situation in terms of processes, structures, and problems, and to identify actors and use cases. These are described for both case studies below.

5.1. Case study 1: the container ship

When a container ship is ready for departure, the first step in leaving the harbour is "letting go the lines" that holds it in a fixed position (Fig. 4) and heaving these lines onboard the ship.

Fig. 4. The aft mooring of Sally Maersk.

While this might seem a trivial procedure, it is in fact both complex and dangerous. As the physical space inside harbours is quite restricted and means for precisely manoeuvring large ships are limited, all lines cannot simply be released simultaneously. Furthermore, when a line is let go, it will remain in the water for a period of time during which no means of propulsion is available due to the risk of lines getting sucked under water and wrapped around the propeller or into the thrusters. Instead the ship can be pulled ahead or astern by means of operating the winches on the remaining lines. Due to these premises, mooring lines are released sequentially in accordance to the need for manoeuvring in a given situation, bringing the ship outside wards and then going ahead when clear of the quay and other ships nearby (illustrated in Fig. 5).

Due to the huge size of container ships, the work tasks involved when letting go the lines are distributed among a number of co-workers located at strategic positions (see Fig. 2). On the bridge (1) chief officers control the rudder, propeller and thrusters; and fore and aft (2 and 3) the first and second officers control the winches for heaving in the lines. Ashore, two teams of assistants lift the lines off the bollards. To insure the safety of the operation, individual work tasks are carefully coordinated and carried out under strict command of the captain. Communication between co-workers in the maritime domain is primarily spoken. While people on the bridge can see and hear each other directly, personnel on deck are, however, out of direct visual and audio contact, and communicate via walkie-talkies.
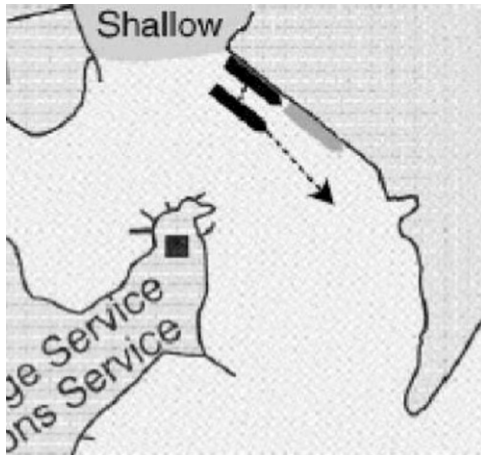
Fig. 5. Sally Maersk leaving Felixstowe harbour.

In order to carry out the operation of letting go the lines in a safe manner, the captain needs overview and control over the propulsion, direction and mooring of the ship. While information about the rudder, propeller and thrusters are available on dedicated instruments on the bridge no information is available about mooring. This only exists as a mental model in the head of the captain. As this mental model is highly sensitive to errors or misunderstandings in the ongoing communication between bridge and deck, and since disparity between the captain's mental model and the real world may cause wrong decisions to be made, considerable resources are spent on establishing and maintaining common ground (Clark and Schaefer, 1989) among the distributed co-workers.

5.1.1. Observed communication problems

Supporting operations on board the container ship, well-established rules and procedures exist for oral communication such as confirming status reports and commands by repeating them back to their sender. To a large extend these procedures work very well. However, from the observations and interviews a number of key limitations emerged related to the use of spoken communication for coordination of collaborative work activities: Sound quality is often poor, communication is not persistent but is time consuming and suffers from bottlenecks due to multiple parallel treads.

Overcoming or reducing (at least some of) these limitations served as an overall motivation for the design of the Maritime Communicator. Inspired by chat applications, newsgroups and Short Messaging Service (SMS) it was our belief that shifting to text-based communication on mobile devices would provide an asynchronous channel making communication persistent and requiring low cognitive overhead (Churchill and Bly, 1999; Popolov et al., 2000).

5.1.2. Communication structures

In order to understand the communication happening while letting go the lines better, video recordings from the bridge of Sally Maersk during multiple instances of manoeuvring inside harbours were transcribed, coded and analysed.

This was used to derive the use cases involved with coordinating and carrying out the operation in terms of the sequence of commands issued by the captain and the structures of subsequent communication between bridge and deck. Furthermore, a complete set of utterances: commands, confirmations and status reports, necessary for coordinating the whole operation was produced. Analysing the typical sequences in the transcriptions of commands issued by the captain also revealed that some commands would always precede others. Also, the possible next commands at any point of the operation could always be deduced. Not surprisingly, this structure was found to correspond to the basics of the "conversation for action" model in speech–act theory (Winograd and Flores, 1986). Hence we realized that if parts of this structure were modelled in the system, the current stage of each step of the operation could be formalized and, for example, represented graphically or integrated with other computer-based data. Also, the potential next stages of any step of the operation could be identified, and possible utterances by any of the communicating actors be deduced, and possibly prioritized.

5.2. Case study 2: the power plant

The power plant is divided into two separate production plants (locations #7 in Fig. 6). The fuel for the two plants is supplied from a large central coal storage area (locations #2 and #3) (Fig. 7).

The fuel department is responsible for delivering the coal used in the two production plants, amounting daily to 5000 t of coal for each. The employees in the fuel department continuously monitor and control the transportation of coal and must ensure that the correct amount of coal arrives at the correct location, and that the coal has certain properties and quality. In order to ensure this, the coal is filtered and grinded (locations #4 and #5). After the coal is processed, it is transported to the two production plants (location #7) by means of underground conveyer belts.
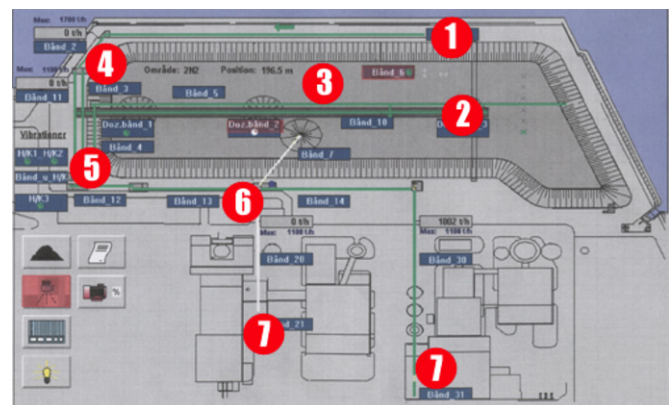

Fig. 6. Key locations of the fuel department.

Another important task for the workers in the fuel department is to prevent the coal from self-combusting in the storage area.

5.2.1. Communication to support coordination

The workers perform a variety of different tasks to ensure that the amount of coal needed is delivered to the two production plants. In order to coordinate the many tasks described above, 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. Communication happens via VHF radios, DECT wireless phones, and sometimes mobile phones. Currently, the control tower (location #6) is the only place where all necessary information is accessible. It is also the place where workers operate and control most of the machinery.

The overall operation of coal transport can be controlled by means of a computerized process control system. However, when a problem occurs that cannot be solved from the control room, for example in the grinder building (location #5), the personnel sent to the site to solve the problem do not have direct access to the control system. Conversely, the specific parts of the individual machinery distributed throughout the plant (for example physical handles on the grinder) can only be operated on site. Consequently, full control of the plant requires communication and coordination between personnel on site and in the control room (Fig. 8).

Fig. 7. Coal storage area.

5.2.2. Observed communication problems

Although DECT phones, VHF radios, and mobile phones offer flexibility and portability, several problems were reported and observed. Firstly, since the conveyor belts run underground and many machines are located inside solid concrete buildings, radio communication is not always reliable due to lack of signal strength. Secondly, there is typically a deafening noise in the tunnels under the plant and inside the buildings, which makes talking to each other difficult and the use of any kind of mobile device for verbal communication virtually impossible. In summary, the workers experience communication problems related to three issues. One is loss of signal, another is noise, and a third is lack of information access on the ground.

In terms of communication structures, similar sequences and confirmation procedures were observed as on board the container ships. This supported our belief that a text-based communication device might also be of value here. It was, however, observed that due to a larger number of locations and objects (i.e. machinery) a wider vocabulary of nouns was used in the operation of the fuel department than in the operation of letting go the lines. The vocabulary of verbs was, however, similarly small, and the same verbs were used in relation to a lot of different machinery (e.g. start, stop, reverse, assist, etc.).

5.3. What did we learn from the two application domain analyses?

The two application domain analyses provided a rich, yet structured, understanding of the use context that we were dealing with. Grounding the analyses in ethnographic field studies meant that a lot of detail was captured, and that our initial design ideas were directly inspired by the use context.

In terms of the application domain analyses there was no notable difference between the two case studies. Both mapped out the overall use context, specific work tasks, means and procedures of coordination, revealed problems related to the use of spoken communication, and discovered structures in communication.

The application domain analyses provided input for the problem domain analysis phase in the form of focus and data, for the design and implementation phase in the form

Fig. 8. Coordination between workers in the control tower and on the ground using VHF radios and mobile phones.

of system context, and for the usability evaluations in the form of information for recreating realistic use contexts and developing realistic and relevant tasks.

6. Problem domain analysis

At this point in the two case studies, we had identified specific work areas of focus, overall division of work, and sequences and structures in communication. As is often the case in projects like these, we had also already begun thinking about design of new technology solutions for the work activities and problems observed. However, rather than moving straight into the design phase, we conducted a second round of analysis as prescribed by Mathiassen et al.'s (2000) OOA&D method. The second round of analysis focused on the problem domain, defined as "the part of the context that is administrated, monitored, or controlled by a system" (Mathiassen et al., 2000). The aim of the problem domain analysis is to support later design and implementation by identifying objects, classes and relations, and providing detailed descriptions of the states that these objects go through. The outcome of the analysis is a description of these objects, classes, structures, and behaviours. The outcomes of the two problem domain analyses are described below, including some of their implications for the design of the resulting prototype systems.

6.1. Case study 1: the container ship

The two major activities of the problem domain analysis for the container ship case study were the modelling of a class diagram for the problem domain (Fig. 9) and a state chart diagram for the *Let go* class.

6.1.1. Class diagram

Modelling the class diagram was a challenge primarily because we extended the problem domain to include additional work activities related to manoeuvring inside the harbour other than the operation of letting go the lines. This was done to support extending the scope of the Maritime Communicator later without having to change the underlying model.

From the class diagram, we identified a number of issues, which directly informed the design of our prototype system.

Firstly, the class diagram illustrated that letting go the lines is one of three similar operations (or tasks) related to manoeuvring inside the harbour. On the basis of this, we found that the Maritime Communicator should have facilities for tailoring an interface supporting each of these specific operations on the basis of a general design for supporting communication and coordination. Thus, general interface elements should be associated to the overall Task class while more specialized interface elements, such as specific graphical representations, should be associated with their appropriate specialization.

Secondly, we found that since the *Commanding officer* class and the *Officer* class aggregate from fundamentally different classes (*Ship* and *Team*, respectively), different interfaces for these actors might also be appropriate. Thus the interface for the commanding officer should possibly include information about the ship while the interface for the officers on deck should possibly include information about the team instead.

Finally the class diagram helped us realize the role of the *Location* class as a mediator between the *Ship* and *Task* classes. Thus knowing the location of a mobile device would enable us to deduce the role of its user in an ongoing task and adapt the interface accordingly.
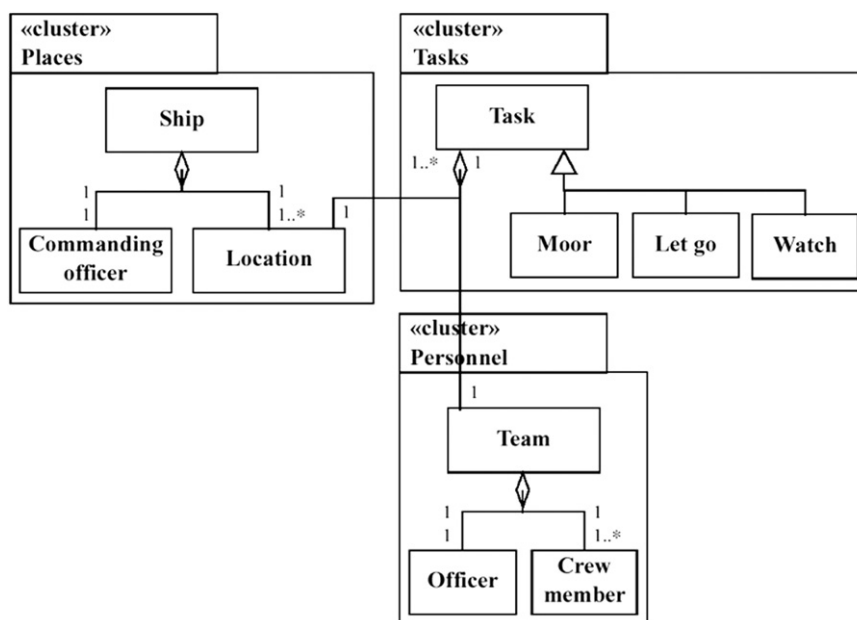

Fig. 9. Class diagram for the problem domain.

6.1.2. State chart diagram

As the *Let go* class would contain the structure of the operation and communication, a lot of effort was put into modelling its state chart diagram (Fig. 10). Even though valuable input for this already existed from the application domain analysis, making a precise model was a challenge. Firstly, a number of variations in the sequence of the operation had to be modelled. Secondly, we had to model

some commands as implicitly containing others, depending on the situation. Finally, we had to model that several commands were being executed in parallel, and could have influence on each other. The state chart diagram was created on the basis of the video recordings and transcripts of communication gathered through the field studies.

The state chart diagram for the *Let go* class provided a detailed view of the operation of letting go the lines. First of all, it provided both the details and abstraction needed to actually implement the idea of partially automating sequence and communication that emerged from the application domain analysis. Secondly, it showed that each command goes through three overall temporal stages of being *imminent, executing,* and *ended.* While in some situations *ended* commands may be important, in other situations only executing commands are vital. From this finding, we were later able to divide the interface into three corresponding areas reflecting enabling a very simple differentiation of priority.

6.2. Case study 2: the power plant

In the power plant case study, we modelled the problem domain in a class diagram that captured the most central objects in the use context of a mobile information and communication system for workers in the fuel department. Informing the development of the class diagram, we also did user profiling in order to describe in more detail the workers whose activities needed to be supported by the system. The outcome of this was the identification of the two primary roles of *controller* and *field worker.*

Providing additional detail for the subsequent design and implementation phase, the problem domain analysis was extended with additional modelling techniques from the related Wisdom method (Nunes and Cunha, 2001a, 2001b). This prescribed the identification of, for example, essential tasks and associated state chart diagrams. Modelling these essential tasks in state chart diagrams, however, turned out to be more difficult because many tasks at the power plant were only carried out rarely, and not all of them had happened during our ethnographic field study. In order to overcome this problem, we decided to intervene and stage a series of situations in which the workers *acted-out* (Howard et al., 2002) work activities, communication and coordination in real world settings. This allowed us to observe these rare situations, adding to the ethnographic data. On the basis of the joint body of data, we identified nine essential tasks that the system needed to support. For each of these tasks, a state chart diagram capturing the flow of the task was produced (see Fig. 11a).

Motivated by the lessons learned from the container ship case study, where the usability evaluation of the functional prototype had later revealed a problem with one of the state chart models from the problem domain analysis, we wanted to validate these models with the users. As we were also interested in some early feedback on a number of design ideas spawned from our analysis, the model validation was
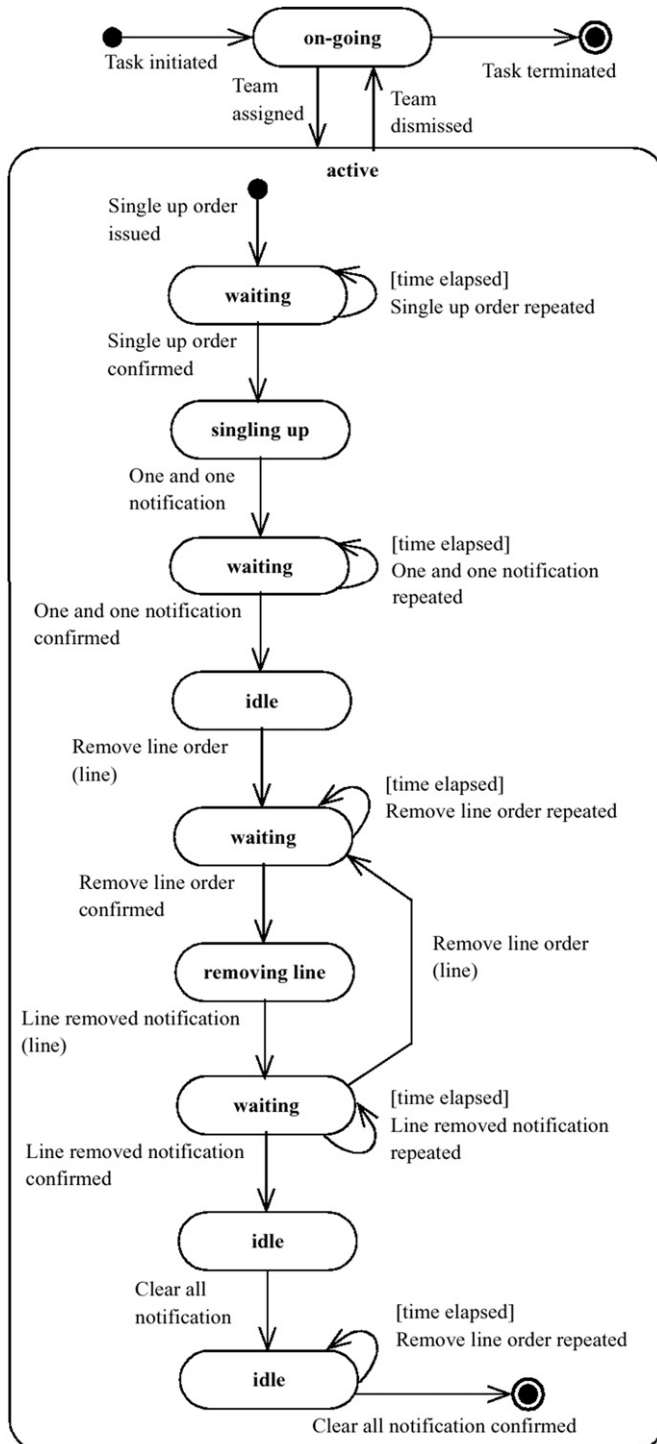

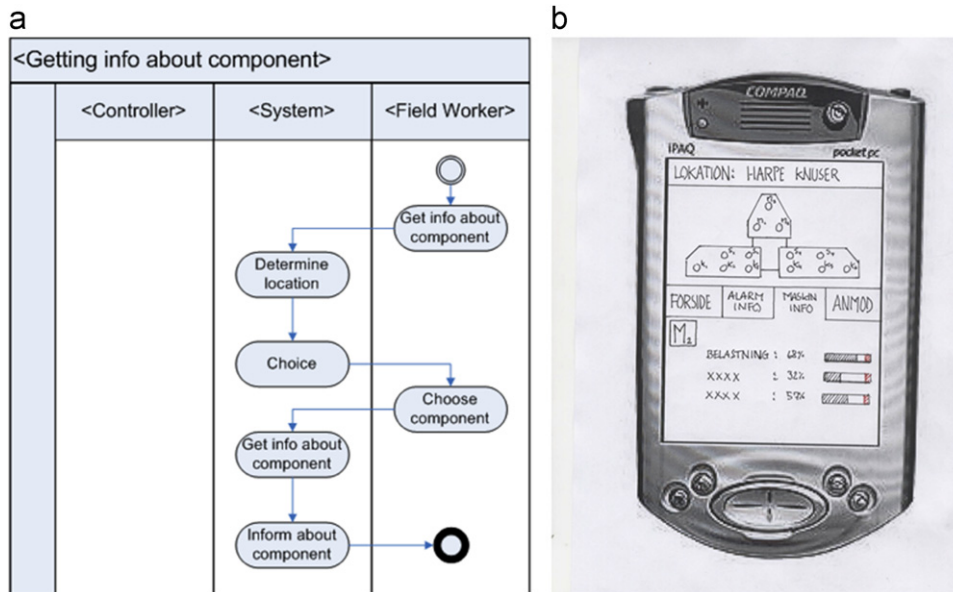Fig. 10. State chart diagram for the *Let go* class.

Fig. 11. State chart diagram for getting information about a component (a) and the corresponding user interface of the first paper prototype (b).

integrated into a paper prototype evaluation (Snyder, 2003). The paper prototype consisted of a series of screens drawn on paper, which could be placed on top of a PDA (Fig. 11b). It was evaluated with real users on site at the fuel department through a series of sessions where a number of workers acted-out their use of the prototype. This provided feedback on both the envisioned functionality and the prototype's structural design, enabling us to evaluate the applicability of our overall design ideas, as well as the underlying models. The evaluation resulted in some modifications to the essential task models, and yielded a number of useful new design ideas. The final class diagram included 9 classes, and the functional list for the system comprised 22 functions.

6.3. What did we learn from the two problem domain analyses?

The two problem domain analyses reduced the complexity of information from the application domain analyses to a series of abstract models, while still preserving the essence of contextual grounding obtained through the ethnographic field study. As described above, creating object-oriented models based on our ethnographic data and application domain analysis not only formalized important aspects of the immediate system context but also generated additional insight into this context than had been gained from the previous phases alone. We learned more about the specific division of labour in the two work domains, the complexities of tasks, and the structures of operations and communication. This additional insight was essential for the subsequent design of our mobile prototype systems.

The two problem domain analyses differed in some specific aspects. While the overall focus and purpose was the same, the second case study deployed additional

techniques from the Wisdom method (Nunes and Cunha, 2001a, 2001b) prompting the development of additional models, for example of essential tasks. This was done to explore the added value of these models when working with ethnographic data and mobile human–computer interaction design. As a result of this addition, we were able to see that some information was missing from our ethnographic field studies, and go back and gather additional data at a relatively early stage of the development process.

7. Design and implementation

The models created in the problem domain analysis formed the basis for the design and implementation phases of the two case studies.

7.1. Case study 1: the container ship

Following the analysis phases, a team consisting of the authors of this article engaged in the design and implementation of the Maritime Communicator. The design and implementation process consisted of four iterations each producing design documents on different levels of abstraction (Fig. 12). Apart from being directly informed by the application and problem domain analyses, the designs produced were inspired by related literature on text-based communication such as Popolov et al. (2000), Smith et al. (2000), Smith and Fiore (2001) and related systems such as chat, newsgroups and the Short Messaging Service (SMS).

7.1.1. Sketches on paper

First, a series of paper based sketches of possible interface designs were produced on the basis of the analysis and inspired by the use of multi-threaded communication
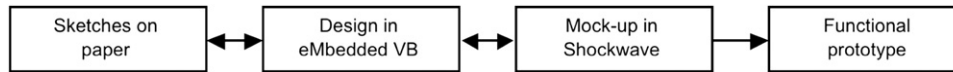
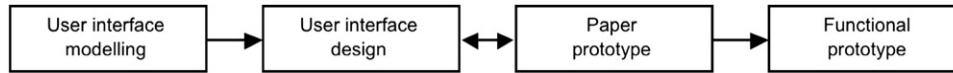Fig. 12. The four stages of our 1st design and implementation phase.

Fig. 13. The four stages of our 2nd design and implementation phase.

and pre-defined messages in other systems. These papers mockup facilitated fundamental discussions of the basic interface design and led to an overall concept providing: (1) a graphical representation of the ship and its mooring lines; (2) a transcription of the communication sorted by objects of reference; and (3) facilitating communication through pre-defined textual commands selected from lists.

7.1.2. Design in eMbedded Visual Basic

On the basis of the paper sketches, detailed design was produced in Microsoft eMbedded Visual Basic. This forced the design team to work within the limitations of the target platform in terms of screen size and graphical interface elements supported in the development toolkit. Consequently, a number of shortcomings arose related to the specific division of screen real estate and the desired level of detail of graphical representations. While most of the subsequent refinements of the design were done directly in Visual Basic, larger design issues such as how to support the textual representation of multiple parallel threads of commands, in what turned out to be a *very* limited graphical space, temporarily forced the design team back to pen and paper.

7.1.3. Mock-up in shockwave

Screen exports from eMbedded Visual Basic (modified in Adobe Photoshop) were used to produce a Shockwave-based mock-up in Macromedia Director showing a possible sequence of communication. Adding life to the Visual Basic-based design in this "quick-and-dirty" fashion facilitated further discussions within the development team and resulted in minor design modifications before doing any real programming. For example, it was decided that the user should not be prompted for confirmations through pop-up screens (as these would temporarily cover vital information). Instead, confirmations were included in the existing list of possible utterances.

7.1.4. Functional prototype

After a final design had been agreed upon, this was implemented in a functional prototype using Microsoft eMbedded Visual Basic. First, the underlying data structure of the system was implemented in accordance with the class diagram and state chart diagrams produced in the

object-oriented analysis. Following this, the user interface was implemented in accordance to the final design. Once the user interface and interaction worked properly we implemented network interfaces and protocols for distributing the system on multiple devices. Implementation time amounted to a total of 15 person-days.

7.2. Case study 2: the power plant

The design and implementation phase for the Power Plant Communicator followed a slightly different process (Fig. 13). System design and implementation was delegated to a group of four developers under the management of the authors, and the design of the Maritime Communicator, along with evaluation data, was provided as a starting point. Instead of design sketching we included an additional object-oriented modelling activity based on the Bridge method (Dayton et al., 1998), and used this activity to direct our specific user interface design.

7.2.1. User interface modelling

As indicated by its name, the Bridge method *bridges* between problem domain analysis and user interface design. This is done through the development of three related models of the user interface on different levels of abstraction: (1) *interaction model,* (2) *presentation model,* and (3) *dialogue model.*

The interaction model describes the interaction spaces and tasks that the system should support. An interaction space is an abstract user interface element, which still has no specific graphical properties. A task is an activity that is carried out by a user employing one or more interaction spaces (Fig. 14).

The left column of the interaction model contains the interaction spaces. The second column contains the tasks, and the three columns on the right represent internal elements of the system.

The presentation model describes each interaction space in more detail, specifying on an abstract level what each individual screen should contain in terms of output elements, input elements and possible user actions but without saying anything concrete about what these elements should look like (Fig. 15).
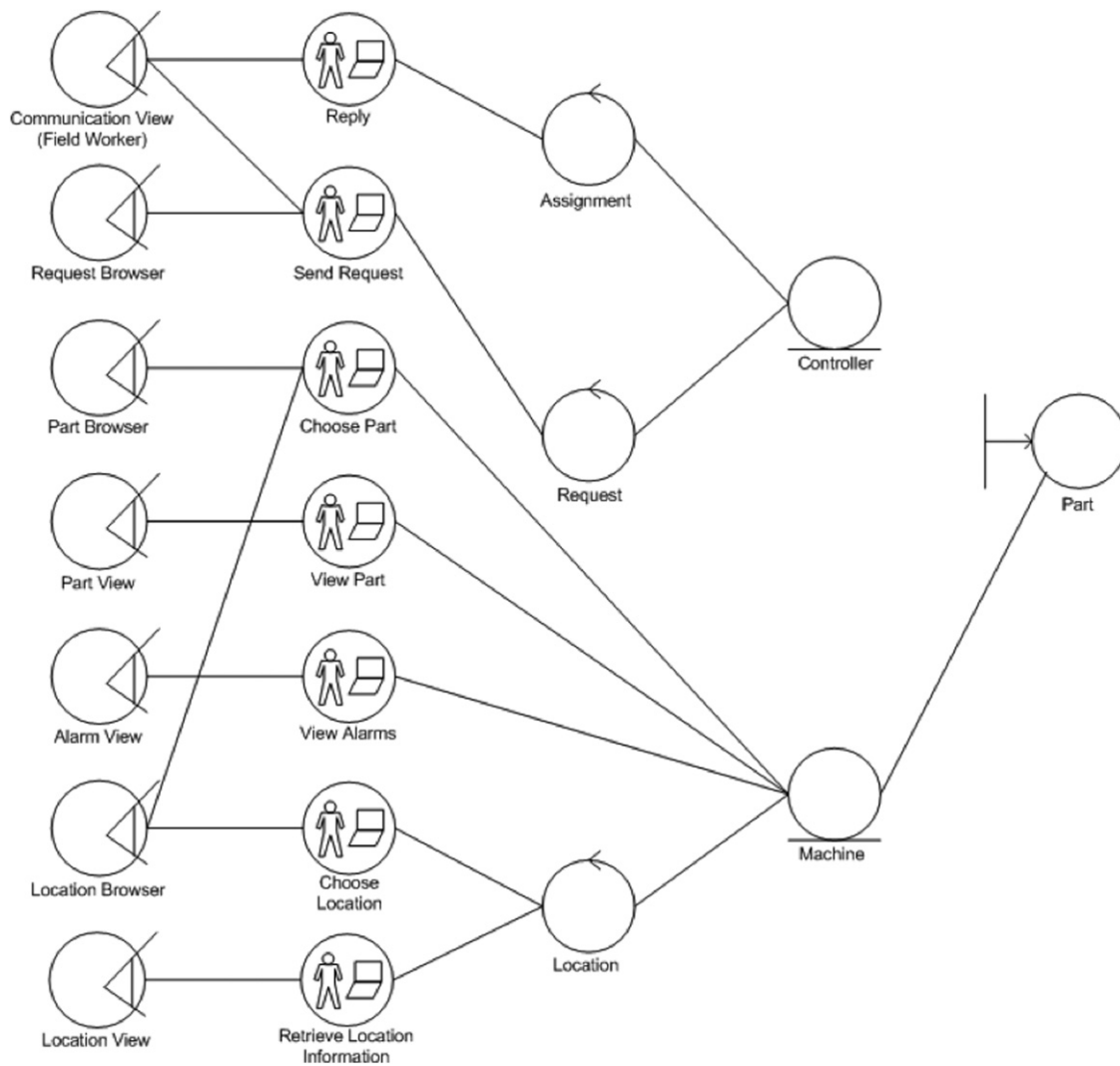
Fig. 14. Interaction model for the Power Plant Communicator.

The attributes of the presentation model classes are defined on the basis of the class diagram for the problem domain. The operations are defined by distributing the function list developed in the problem domain analysis between the individual presentation model classes.

7.2.2. User interface design

The first step of the user interface design activity was to transform the interaction, presentation and dialogue models into concrete design. Informing this process, the three models from the Bridge method described above provided important input on different levels. The interaction model provided input on the overall structure and relationship between individual screens. The presentation model provided a detailed list of elements that we had to include on each specific screen. The dialogue model described the sequence by which the workers would typically interact with the screens.

As illustrated in Fig. 16, knowing what information and functionality to provide, and when, proved to be highly valuable information in order to maximize the use of the small screen real estate of the target platform. In order to reduce complexity, Fig. 16 only depicts some of the connections between interaction spaces and screen design.

7.2.3. Paper prototype

The user interface design developed on the basis of the interaction, presentation and dialogue models was transformed into a second paper prototype with a high level of fidelity (Fig. 16 right). The paper prototype was then evaluated in situ at the power plant with prospective users acting-out a series of realistic use scenarios. Observations from the paper prototype evaluations led to refinements of the user interface design.

7.2.4. Functional prototype

Following the paper prototype evaluations, a functional prototype that matched the final design specifications was implemented in C# using Microsoft Visual Studio .Net 2003 Professional and the .Net Compact Framework.
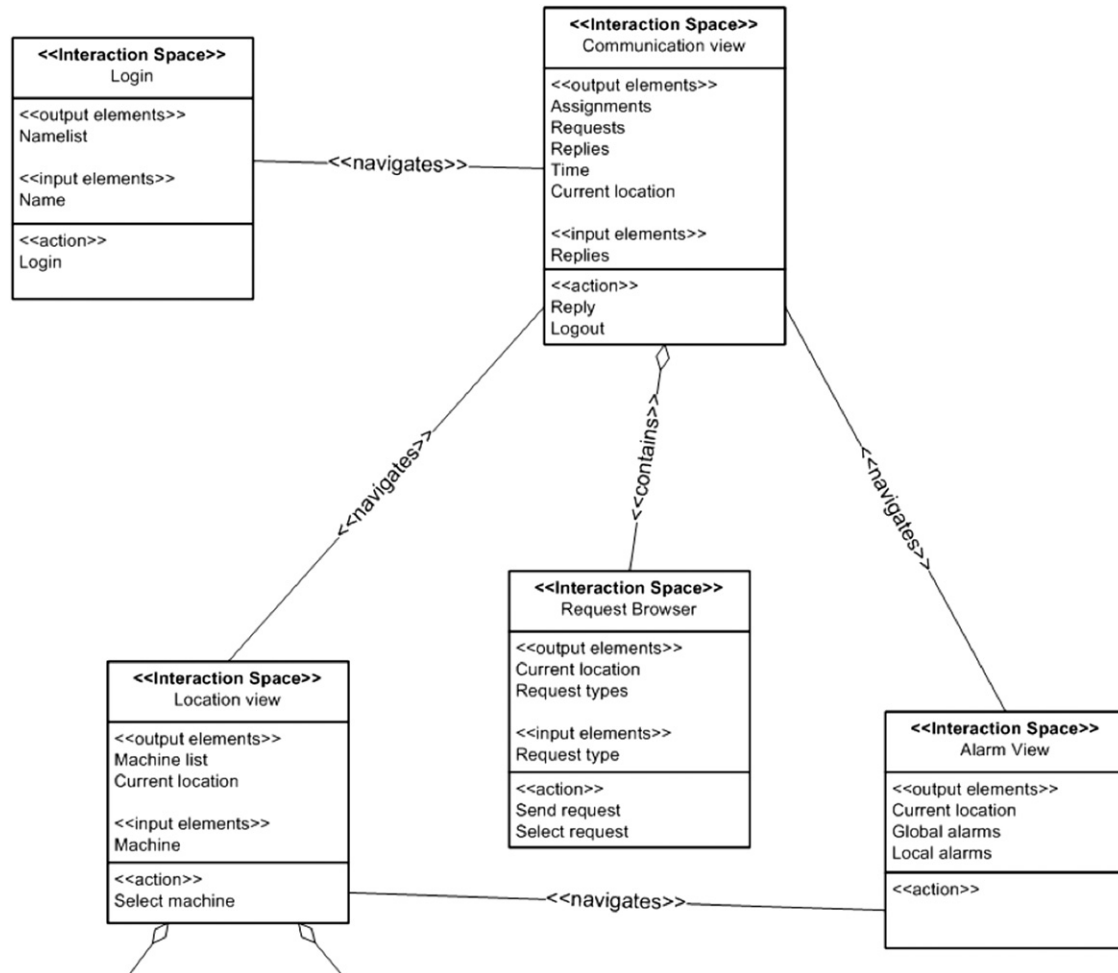
Fig. 15. Part of the presentation model for the Power Plant Communicator.

The implementation activity amounted to a total of 20 person-days.

7.3. What did we learn from the two design and implementation processes?

While the two implementation activities were very similar in process and the amount of time spent, the design activities differed notably from case study 1 to case study 2. The differences reflect that the design activity was more exploratory in nature in the first case study and more structured in the second one. Both processes have their strengths and weaknesses and are applicable depending on the nature of the specific project. The design process for the Maritime Communicator represents an iterative approach where design is carried out with an increasing level of detail using different means of expression such as paper, interface toolkits, and mock-ups. In this approach, the diagrams produced in the problem domain analysis were important for reference, but design was very much driven by creativity. The design process of the Power Plant Communicator, on the other hand, represents a more structured approach where the outcomes from initial analyses are directly elaborated on in the form of

additional models defining abstract properties of the user interface. The "creative" part of the design phase is therefore postponed for longer but in effect informed more directly by ethnographic field study through the models developed. In our experience, the additional object-oriented models focussing explicitly on the user interface helped in keeping the design of the Power Plant Communicator grounded in our field data. In particular, we found that the Interaction Space models provided structured input into the design of each screen of the system without prescribing their look and feel in a way that impeded creativity.

The second major difference between the two design phases was that case study 2 involved user feedback on a paper prototype by subsequent field visits to the context of use prior to implementation. Again, this helped us maintain the grounding of our design process in the initial ethnographic field study to a stronger degree than in case study 1.

8. Functional prototypes

In this section we describe the two functional mobile device prototypes resulting from the field study, analysis, design, and implementation processes described above: the

Fig. 16. Mapping of presentation model classes to user interface design.

Maritime Communicator and the Power Plant Communicator. The prototype systems are not the contribution of this article but serve as illustrations of outcome. The two systems are described in detail in Kjeldskov and Stage (2006) and Kjeldskov et al. (2006).

8.1. The Maritime Communicator

The Handheld Communicator was targeted at PDAs running the Microsoft PocketPC operating system such as the industrial grade Symbol PPT8800 series (Fig. 17). Apart from a touch screen, such industrial devices support interaction by means of large rubber buttons located below the display and suitable for one-handed interaction. Due to the potentially harsh conditions of use, in which pen-based interaction might be problematic, all interaction took place through these buttons.

The system set-up consisted of three PDAs connected through a wireless network. One device was intended for the captain on the bridge while the other two were for the 1st and 2nd officers on the fore and aft deck, respectively.

8.1.1. Overall design

The Maritime Communicator explores the ideas described in Section 5 replacing verbal communication with an exchange of pre-defined text messages grouped by objects and the state of tasks, and providing an updated representation of the shared work domain.

Fig. 17 shows the user interface of the Handheld Communicator for the captain. At the bottom of the screen there is a list of unexecuted commands and confirmations. The order of the list corresponds to the standard sequence of the overall operation, and possible utterances only appear when appropriate in relation to the state of the task. By default, the most likely next step of the operation is highlighted (based on the state chart diagrams).

The most important element of the interface is the list of "ongoing tasks". When a command is executed, it appears on this list along with a counter displaying the time passed since it appeared. When a command is confirmed the timer is substituted by the text "[ok]" followed by a description of the current activity (e.g. "Singling up…"). When a task is reported completed, a short statement (e.g. "1 and 1 fore") substitutes the description of activity and the captain is prompted for confirmation. When the captain confirms the completion of a task, the communication thread is moved to the "history" list. When the history list is full, it scrolls the oldest commands and statements out of sight. At the top of the screen a simple pictogram displays the current state of the operation for quick reference: the lines still attached to the quay and the current status of fore and aft.

On deck, the interface for the officers is very similar to that on the bridge thus providing a view on the present status of the mooring and a list of all past and ongoing communication among the distributed co-workers.

Fig. 17. The Maritime Communicator on a Symbol PPT8000.

The main difference is the list of specific commands and utterances available, which is filtered on the basis of their location and task, and the indications under "ongoing tasks" of commands with responses pending.

8.1.2. Implementation

The application running on the captain's device worked as a server containing a formalized representation of the communication pattern of the task. The devices on deck logged on to this server and identified their physical location. During operation, function calls and unique command identifiers were exchanged over the wireless network. All network traffic was broadcast but processed and represented differently on each device in accordance to their physical location (bridge, fore or aft).

8.2. Case study 2: the power plant

Like the Maritime Communicator, the Power Plant Communicator was targeted at industrial grade PDAs running the Microsoft PocketPC operating system. Interaction was facilitated by the device's function keys and through finger-based input on the touch screen. In addition to the handheld terminals used by the workers moving around the plant, a desktop PC interface was designed for use in the control tower.

8.2.1. Overall design

The Power Plant Communicator provides mobile distributed workers with access to information in the central computer system about the general status of the plant and about the specific machinery within their proximity. It also gives them a simple text-based communication channel for coordinating certain work activities. The system is divided into three overall screens with a number of associated sub-screens: (1) Communication screen, (2) Alarm screen, and (3) Status screen. At the top of the display, the system indicates who is logged in, where the user is located and what time it is. This information is important for the field workers because it gives them a frame of reference for interpreting the information and functionality provided by the system.

8.2.2. The communication screen

The communication screen (Fig. 18) provides workers with a text-based communication channel (inspired by the Maritime Communicator). Using the two lists next to each other above the panes the user can compose a message by combining a verb with a noun, for example "stop production". Above these two lists there are three buttons for sending the composed message, or for sending a standard reply that you Accept (Accepter) or Reject (Afvis) a particular request. To avoid pressing the wrong button by mistake, the buttons for accepting or rejecting are placed furthest apart. Above these three buttons the ongoing conversation is displayed on a list. The list is divided into a series of conversation threads, grouping communication about the same object or task together. To make a clear difference between requests and confirmations, the latter are indented and have their first word (for example ACCEPT or REJECT) in capital letters. To add a new message to a thread the user selects an utterance on the list, for example "Check Harper/Knuser 2, and sends a response.
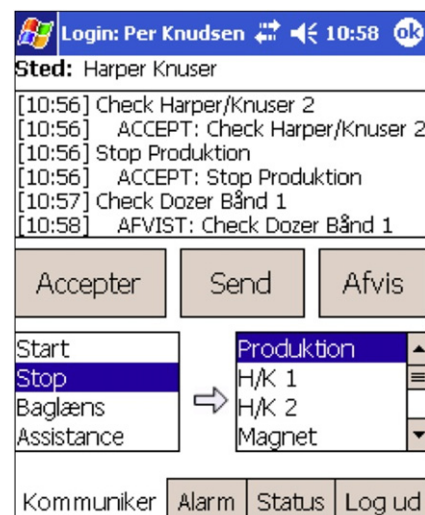

Fig. 18. Communication screen.

A main difference in this design compared to the Maritime Communicator is that messages are composed from two lists rather than one. This design emerged from the object-oriented models revealing that the number of possible utterances was very large, but that they were all composed from a common set of verbs and a set of nouns related to the users' location. Because the system knows where the workers are physically located at the power plant, it can deduce which machinery they are most likely to be communicating about (generating the list of nouns). Knowing about the functionality of the machinery the system can deduce what actions the user can request (generating the list of verbs).

8.2.3. The alarm screen

The alarm screen (Fig. 19) alerts the workers if something is wrong at the power plant and provides them with the available details about the problem. If an alarm is associated with the users location, detailed information is displayed immediately. The alarm screen contains a graphical representation of the machinery in question, with the area of the problem highlighted in red. Below the graphical representation, there are two lists displaying alarms at the users current location and other alarms along the production line. These lists were included because our models showed that problems along the production line often cause other problems to happen, requiring co-located workers to start coordinating a shared strategy. Using these lists, the workers can get an overview of the cause and/or effects of a specific problem, thus assisting them in assessing its criticality. Each alarm has a timestamp that enables the workers to determine their sequence and current relevance.

8.2.4. The status screen

The status screen (Fig. 20) provides workers with access to information about machinery in the production line
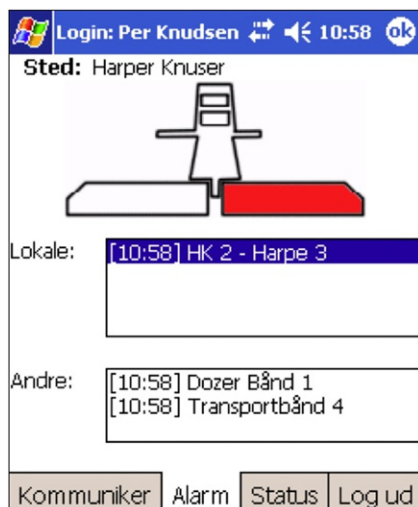

Fig. 19. Alarm screen. (For interpretation of the references to colour in this figure, the reader is referred to the web version of this article.)
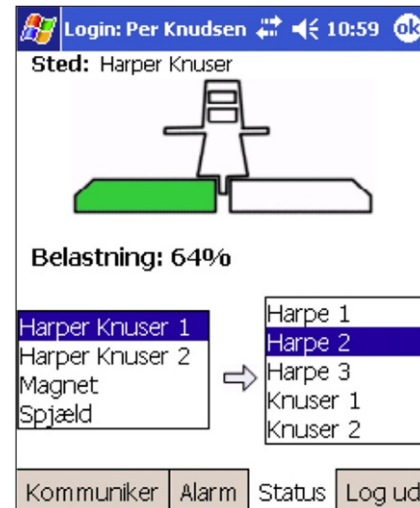

Fig. 20. Status screen. (For interpretation of the references to colour in this figure, the reader is referred to the web version of this article.)

within their close physical proximity. Below the graphical representation, the users can choose which of the machines within proximity they want to work with (the list on the left), and what specific part of it they want to access information about (the list on the right). The screen can also function as a remote control to some machines basic operations such as start, stop and reverse. The status screen looks similar to the alarm screen in many ways. At the top of the screen, there is a graphical representation of the machine being accessed, with the specific part chosen being highlighted. If the chosen part of the machine is functioning or the load on the part is at a normal level, it is highlighted in green. If the load climbs towards a critical level, the colour changes to orange. If reaching a critical level, the colour changes to red and an alarm is activated. These states were revealed through our models.

8.2.5. Implementation

The technical implementation of the Power Plant Communicator was similar to the Maritime Communicator. The computer in the control tower (a desktop PC) worked as a server containing a formalized representation of the power plant and typical work activities. The handheld terminals logged on to this server and identified their physical location, following which an appropriate interface was displayed on them. During use, function calls and commands were exchanged over the network, using handshake to confirm delivery, thus eliminating commands being "lost in the air". All network traffic was broadcast but processed and represented differently on each device in respect to their location.

9. Evaluations and user feedback

The final phase of our two case studies was to evaluate the usability of the produced prototype systems. In the context of this article the evaluations below serve as

validations of the outcome of the development processes as well as illustrations of different approaches to "closing the circle" by going back to the domains where our ethnographic field studies were originally conducted. Due to the nature of the two use domains, both evaluations were challenging to carry out in a realistic, but safe, manner. This challenge was addressed in two different ways: by carrying out the evaluation in a high-fidelity ship simulator and by acting out use scenarios at the power plant.

9.1. Case study 1: the container ship

The Maritime Communicator was evaluated in a state-of-the-art ship simulator at Svendborg International Maritime Academy in Denmark (Kjeldskov and Skov, 2007). The evaluation involved three teams of two experienced maritime officers as test subjects. The officers were given the task of letting go the lines and departing from harbour using the Maritime Communicator for communication and coordination. One subject user acted as captain on the bridge (Fig. 21) while the other acted as 1st officer on the fore deck in a neighbouring room.

The ship simulator was set up to imitate the operation of a large ship in challenging weather and traffic conditions. The scenario was developed in collaboration with the simulator division of Svendborg International Maritime Academy, and corresponded to a real world situation observed during our field studies described above in Section 5 (see Fig. 5).

During the evaluation the captain had to consider all aspects of manoeuvring the ship. This included controlling the rudder, propellers and thrusters as well as communicating with personnel on the ship, harbour traffic control and taking into consideration the movements of other ships. The 1st officer on deck verbally forwarded commands to his team of assistants on the quay and manning the winches and reported progress back to the captain. The team of assistants were a part of the digital simulation.

During the evaluation, the users were asked to think-aloud, explaining their experience of and interaction with

Fig. 21. Evaluation of the Maritime Communicator in a ship simulator.

the prototype. Two evaluators located on the bridge and on the deck, respectively, observed the users and asked questions for clarification. Following each evaluation session, a group interview of 10–15 min was carried out. The evaluation sessions were recorded on digital video from four camera sources and audio capturing close-up views of the handheld devices and overall views of the use context.

9.1.1. Highlights from the evaluation

The evaluation of the Maritime Communicator in the ship simulator provided rich data on the usability of the design. First of all, the user study showed that the prototype could actually replace the majority of spoken communication between the captain and officers. Also, the users expressed that the text-based interface gave them a simple channel of communication in which they could easily monitor the ongoing communication and progress of the operation. As a supplement to this, some officers on deck expressed a wish for having commands read out by a synthetic voice as an option when they were busy with their hands elsewhere. Generally, the users learned what to do and what feedback to expect within the completion of one or two threads of communication and reported that the design was very intuitive to use. The differentiation between commands appeared straightforward as did the grouping and progress of ongoing threads. The graphical representation of the ship was highly appreciated for overview, as this was not currently provided.

Apart from the positive feedback, the user study also revealed 22 usability problems experienced by the users. First of all, we identified a need for being able to request or report something out of the ordinary, and to correct or withdraw an utterance. This finding indicated that our ethnographic field study had missed situations where there was a need to communicate things out of the ordinary, and consequently this was also missing from our problem domain analysis. This lesson learned led us to modify the analysis procedure in the subsequent Power Plant Communicator case study to involve a validation of these models, facilitated by an initial paper prototype evaluation, after the problem domain analysis, as described earlier.

9.2. Case study 2: the power plant

The Power Plant Communicator was evaluated in situ at the fuel department. The evaluation involved five workers with several years of work experience with this particular plant. The prototype system was used during live operation of the power plant while carrying out a series of typical work tasks. In addition to this, a series of less common work activities were acted-out in situ as we had done with the evaluation of the paper prototypes described earlier.

One researcher managed the evaluation and asked questions for clarification. Another researcher recorded

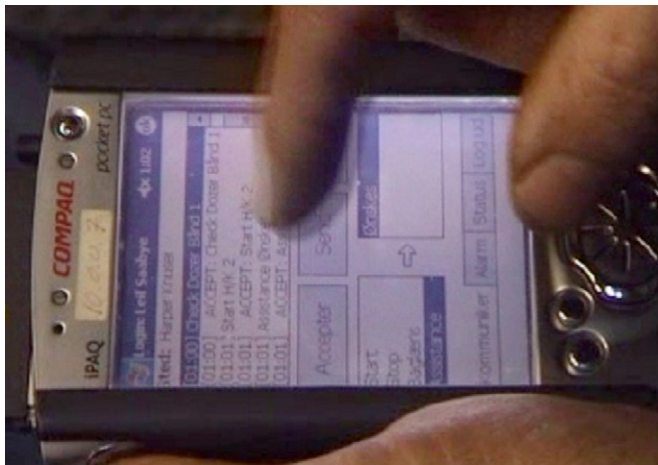Fig. 22. Field evaluation of the Power Plant Communicator in situ.

Fig. 23. Close-up of the communicator during field evaluation.

the evaluation on a handheld video camera shifting focus between the workers, the settings, and the screen on the handheld device (Figs. 22 and 23).

The users were asked to think-aloud, but because of the noise level in some buildings, this did not always work well. In these situations, a post-use interview was conducted outside the building immediately afterwards.

9.2.1. Highlights from the evaluation

The results of the evaluation showed us that the system was indeed usable and that the workers were satisfied with using it. The users had no problems understanding and adopting the basic functionality of the system. Some compared it to sending SMS messages on their mobile phone. Others compared it to a remote control. In relation to the communication screen, the users reported that they liked to be able to combine text-based messages and spoken communication over the VHF radio. Text-based communication was primarily found useful when noise prevented spoken commands but was also reported very useful in complex situations where they otherwise needed to remember what had been said when and by whom.

Regarding the alarm screen, the users reported that the annotated graphical representation would help them greatly in locating and fixing problems quickly. Some reported that they would like even more detailed graphics and preferably a plant overview as well. On the status screen the users found the simple access to information about machines and control of it within physical proximity compelling and highly useful for their daily operation of the plant. Furthermore, not being able to operate machinery out of proximity was perceived as a significant safety advantage. Apart from the positive observations and feedback the evaluation also revealed 14 usability problems experienced by the users, to be addressed in the next iteration of design and development.

10. Summary and discussion

The challenge we set out to respond to by combining ethnography with Mathiassen et al.'s (2000) object-oriented analysis method was that mobile software development requires insight into rich and dynamic use contexts, but that ethnographically based approaches, which are good for studying such contexts, provide data and insights that can be hard to translate into system design. This led us to devise an approach that explicitly combines ethnography with a structured analysis and design approach, enabling us to obtain both contextual richness and abstract models for informing the development of mobile systems. In the following, we summarize and discuss some of the lessons learned from the two case studies of using this approach presented above.

10.1. Value of application domain analysis

The primary value of applying an object-oriented application domain analysis to the ethnographic field data was that this approach provided a specific focus for the data. Rather than doing an open-ended or grounded type of analysis, which outcomes can be difficult to translate into a system development process, the application domain analysis approach prompted us to specifically address the question of what the system is going to be about, and to generate a set of overall requirements for what we were going to build. Apart from the obvious value of these tangible outcomes, the value of the application domain analysis also lay in the process of doing these analyses as a

shared activity within the project teams. This prompted focused discussions about what was observed in the field, and through these discussions we gradually understood more and more the domains we were working with, and were essentially developing systems for. As pointed out above, the outcome of this analysis was able to serve as a common frame of reference for the project teams throughout the rest of the project, representing one of the outcomes from the ethnographic field study.

10.2. Value of problem domain analysis

The two object-oriented problem domain analyses reduced the complexity of information from the application domain analyses to a series of abstract models, while still preserving the essence of contextual grounding obtained through the ethnographic field study. It was at the time of the problem domain analysis that the added value of the object-oriented approach particularly began showing its strengths for the process of system development by prompting the development teams to formalize structures and processes observed in the ethnographies into abstract models. During this process we learned more about the specific division of labour in the two work domains, the complexities of tasks, and the structures of operations and communication than had been obvious from the ethnographies. This additional insight was essential for the subsequent design processes, and the models created were also essential for the later implementation phases. In the power plant case a particular valuable experience with the problem domain analysis was that this activity revealed that some information was missing from our ethnographic field studies, and made us go back and gather this additional field data at a relatively early stage of the development process (as discussed more below). This exemplifies how structured modelling can act as a verification of the scope of the much more unstructured ethnographic study, and illustrates a strength of combining these two approaches in system design.

10.3. Informing the design process

The value of the object-oriented approach for the design process was slightly different for the two case studies. The design process of the Maritime Communicator represented an approach where the diagrams produced in the problem domain analysis were important as structural reference for an iterative and highly creative design activity. By comparison, the design process of the Power Plant Communicator represented an approach where outcomes from the object-oriented analyses were directly elaborated on through additional abstract modelling before any concrete design activity was carried out. Although this meant that actual interface design was done later in the process, this was at the same time informed more directly by the ethnographic field study. By developing object-oriented models focussing explicitly on the structural design of the user interface,

such as Interaction Space models, the design of the Power Plant Communicator remained strongly grounded in data from the ethnography.

10.4. Supporting the implementation process

The value of the object-oriented analysis and design efforts for the implementation of our prototype systems were two-fold. Firstly the structured and iterative approach had informed the creation of a well-specified system and user interface. This minimized effort spent during the implementation phase on essentially doing design, and ensured that focus could be kept on the task of coding. Secondly, and equally importantly, the abstract models developed during the problem domain analysis, and for the power plant case also during the design phase, directly informed the structure of the software programs, and contained all the information required to build the system. This minimized time spent during the implementation phase on essentially translating design specifications into structures that could be coded, and ensured that the resulting code correlated well with the specified design.

10.5. Value of ethnography in mobile HCI

Taking a step back, the ethnographic field studies described in Section 4 played an invaluable role in the two system development processes, confirming the potential value of ethnography in mobile HCI as suggested by Kjeldskov and Graham (2003), and illustrated by, for example, Luff and Heath (1998) in their work on mobility in collaboration back in the 1990s. Both of the studies presented in this paper generated rich insights into the use contexts and work activities that would unlikely have been reached otherwise. Firstly, the ethnography on board the container ships generated insights and inspiration that, amongst others, led to the conception of the text-based communicator systems in the first place. In comparison, the ethnography at the power plant was more focused, but confirmed observations from the maritime domain, and led to the identification of different, yet related work activities where similar technology might be applicable, as well as identification of new, unique type of work activities not observed on board the container ships.

Secondly, the two ethnographies generated valuable data for contextually rich application and problem domain analyses. As can be seen in Sections 5 and 6, these analyses of the general use context, and the specific part of that context that is directly related to the use of the proposed systems, were particular rich in details and strongly grounded in real world observations. This made them extremely useful for the further development process as a frame of reference representing the data from the ethnographies in a tangible and organized form.

Thirdly, the two ethnographies provided insight into the use contexts that was critical for the recreation of realistic test environments for the evaluation of prototype systems.

For both cases this included the creation of realistic and relevant tasks. For the container ship case it also included informing the scenario programmed into the ship simulator.

The only limitation of the ethnographic approach experienced related to the issue of missing out of observing a rare event. This was partly due to the inherent naturalistic and non-intrusive nature of the ethnographic methodology, and partly due to the length of time spent in the field. One approach to overcome this issue is, of course, to spend longer time in the field until all possible events have happened and been observed. However, apart from the intrinsic uncertainty about exactly when this criterion has been reached, this is typically not a realistic approach in a software development project. As a response to this, various time-optimized field study techniques such as "rapid ethnography" (Millen, 2000) and "rapid contextual design" (Holtzblatt et al., 2005) have emerged, which, amongst others, propose that special effort is made to identify and cover those unusual situations also. The next question here is, of course, *how* to do that? As one possible answer, we experienced that the systematic nature of the objected-oriented analysis provided valuable input for testing the completeness of the ethnographic data gathered, and subsequently guiding the focus of final fieldwork. Consequently, it can be argued that ethnography and object-orientation is best combined iteratively, with going back and forward between field studies and analysis being an explicit part of the approach.

11. Conclusions and further work

We have presented two case studies of mobile system development for supporting distributed collaborative work activities within industrial process control. Based on the lessons learned from these two case studies, we have described and discussed how ethnography and object-orientation can be successfully combined for obtaining contextual richness and abstract models for mobile interaction design.

One of the challenges reported is that ethnography generates findings and knowledge with such contextual richness that it can be hard to transfer into system design. By combining ethnography with the software engineering method of object-oriented analysis and design, we have shown how ethnographic field data is a highly valuable source of input for developing object-oriented models by providing contextual richness and that, in turn, objected-oriented analysis is a highly valuable method for working with ethnographic field data in systems development by supporting the creation of abstract models. Combining the two, we have demonstrated a method where the use of ethnographic field studies in system design is supported by the use of a structured analytical approach, and where abstract modelling of systems is informed by rich contextual data. By applying this method, ethnographies are empowered in their capability for informing core

system design, and system design is consequently strongly grounded in their use context.

We have applied the proposed methodological approach to two different mobile system development processes over a period of 5 years. In doing this we found that the experienced value of the method generalized from one case study to the other, and we were even able to make some improvements to the specific analytical activities involved. While on the basis of our experiences we believe that the described process has general value for the development of other mobile systems, this should, however, be evaluated through other cases of mobile interaction design applying the method of explicitly combining ethnography and object-oriented analysis and design.

Acknowledgements

References

Andersen, P.B., May, M., 2001. Tearing up interfaces. In: Liu, K. (Ed.), Information, Organization and Technology. Studies in Organizational Semiotics. Klüwer, Boston.

Bardram, J.E., Hansen, T.R., 2004. The AWARE architecture: supporting context-mediated social awareness in mobile cooperation. In: Proceedings of the CSCW '04. ACM, pp. 192–201.

Benyon, D., 2002. Representations in human–computer systems development. Cognition, Technology & Work 4 (2002), 180–196.

Beyer, H., Holtzblatt, K., 1998. Contextual Design—Defining Customer Centred Systems. Morgan Kaufmann, San Francisco.

Churchill, E.F., Bly, S., 1999. It's all in the words: supporting work activities with lightweight tools. In: Proceedings of the ACM Siggroup'99, pp. 40–49.

Clark, H.H., Schaefer, E.F., 1989. Contributing to discourse. Cognitive Science 13, 259–294.

Constantine, L., Biddle, R., Noble, J., 2003. Usage-centered design and software engineering: models for integration. In: Proceedings of the ICSE 2003. IFIP, Portland, USA, pp. 106–113.

Crabtree, A., Rodden, T., Tolmie, P., Button, G., 2009. Ethnography considered harmful. In: Proceedings of the CHI '09. ACM, pp. 879–888.

Crabtree, A., 2004. Design in the absence of practice: breaching experiments. In: Proceedings of the DIS 2004, Cambridge, MA, USA, pp. 59–68.

Dayton, T., McFarland, A., Kramer, J., 1998. Bridging user needs to object oriented GUI prototype via task object design. In: Wood, L.E.

(Ed.), User Interface Design—Bridging the Gap from User Requirements and Design. CRC Press LLC.

Diggins, T., Tolmie, P., 2003. The 'adequate' design of ethnographic outputs for practice: some explorations of the characteristics of design resources. Personal and Ubiquitous Computing 7, 147–158.

Holtzblatt, K., Wendell, J.B., Wood, S., 2005. Rapid Contextual Design—A How-to Guide to Key Techniques for User Centered Design. Morgan Kaufmann, San Francisco.

Hosbond, J.H., 2005. Mobile systems development: challenges, implications and issues. In: Proceedings of the MOBIS, IFIP TC8, 2005, Leeds, UK.

Hosbond, J.H., Nielsen, P.A., 2005. Mobile systems development—a literature review. In: Proceedings of the IFIP 8.2: Designing Ubiquitous Information Environments. IEEE, Cleveland, Ohio.

Howard, S., Carroll, J., Vetere, F., Murphy, J., Peck, J., 2002. Provoking innovation: acting out with contextual scenarios. In: Proceedings of the HCI 2002, BCS-HCI.

Hughes, J., King, V., Rodden, T., Andersen, H., 1995. The role of ethnography in interactive systems design. Interactions 2 (2), 56–65.

Hughes, J., King, V., Rodden, T., Andersen, H., 1994. Moving out from the control room: ethnography in system design. In: Proceedings of the CSCW '94. ACM, Chapel Hill, USA, pp. 429–439.

Iqbal, R., James, A., Gatward, R., 2005. Designing with ethnography: an integrative approach to CSCW design. Advanced Engineering Informatics 19 (2005), 81–92.

Kjeldskov, J. A longitudinal review of mobile HCI research methods, forthcoming.

Kjeldskov, J., Skov, M.B., 2007. Studying usability in situ: simulating real world phenomena in controlled environments. International Journal of Human–Computer Interaction (IJHCI) 22 (1), 7–37.

Kjeldskov, J., Nielsen, C.M., Overgaard, M., Pedersen, M.B., Stage, J., Stenild, S., 2006. Designing a Mobile Communicator: combining ethnography and object-oriented design. In: Proceedings of the OzCHI 2006. ACM and CHISIG, Sydney, Australia, pp. 95–103.

Kjeldskov, J., Stage, J., 2006. Exploring "canned communication" for coordinating distributed mobile work activities. Interacting with Computers 2006 (18), 1310–1335.

Kjeldskov, J., Stage, J., 2004. New techniques for usability evaluation of mobile systems. International Journal of Human–Computer Studies (IJHCS) 60 (2004), 599–620.

Kjeldskov, J., Graham, C., 2003. A review of mobile HCI research methods. In: Proceedings of the Mobile HCI 2003. Springer-Verlag, pp. 317–335.

Kjeldskov, J., Stage, J., 2003. Designing the handheld Maritime Communicator. In: Proceedings of the First Conference on Designing User Experiences, DUX 2003. ACM, San Francisco, CA, USA, pp. 1–15.

Krogstie, J., Lyytinen, K., Opdahl, A.L., Pernici, B., Siau, K., Smoland, K., 2004. Research areas and challenges for mobile information systems. International Journal of Mobile Communications 2 (3), 220–234.

Luff, P., Heath, C., 1998. Mobility in collaboration. In: Proceedings of the CSCW'98. ACM, Seattle, Washington, pp. 305–314.

Lumsden, J., Langton, N., Kondratova, I., 2008. Evaluating the appropriateness of speech input in marine applications: a field evaluation. In: Proceedings of the Mobile HCI '08. ACM, pp. 343–346.

Mathiassen, L., Munk-Madsen, A., Nielsen, P.A., Stage, J., 2000. Object-Oriented Analysis & Design. Marko Publishing, Aalborg.

Mikkonen, M., Vayrynen, S., Ikonen, V., Heikkila, O., 2002. User and concept studies as tools in developing mobile communication services. Personal and Ubiquitous Computing 6, 113–124.

Millen, D.R., 2000. Rapid ethnography: time deepening strategies for HCI field research. In: Proceedings of the Third Conference on Designing Interactive Systems (DIS '00). ACM, New York, NY, USA, pp. 280–286.

Nielsen, C., Søndergaard, A., 2000. Designing for mobility—an integration approach supporting multiple technologies. In: Proceedings of the NordiCHI 2000, Stockholm, Sweden.

Norman, D., 1990. The 'problem' with automation: inappropriate feedback and interaction not over automation. In: Broadbent, D.E. (Ed.), Human Factors in Hazardous Situations. Clarendon Press, Oxford, pp. 137–145.

Nunes, N.J., Cunha, J.F., 2001a. Wisdom: a software engineering method for small software development companies. IEEE Software.

Nunes, N.J., Cunha, J.F., 2001b. Wisdom—whitewater interactive system development with object models. In: van Harmelen, M. (Ed.), Object Modeling and User Interface Design. Addison-Wesley, New York.

Paay, J., Sterling, L., Vetere, F., Howard, S., Boettcher, A., 2009. Engineering the social: the role of shared artifacts. International Journal of Human–Computer Studies 67, 437–454.

Paay, J., 2008. From ethnography to interface design. In: Lumsden, J. (Ed.), Handbook of Research on User Interface Design and Evaluation for Mobile Technology. Idea Group Inc. (IGI), PA, USA, pp. 1–15.

Popolov, D., Callaghan, M., Luker, P., 2000. Conversation space: visualising multi-threaded conversation. In: Proceedings of the AVI2000. ACM, Palermo, Italy, pp. 246–249.

Rasmussen, J., 1983. Skills, rules and knowledge: signals, signs and symbols and other distinctions in human performance models. IEEE Transactions on Systems, Man and Cybernetics 13, 3.

Rasmussen, J., 1986. Information Processing and Human–Machine Interaction. North-Holland, New York.

Rügge, I., Ruthenbeck, C., Piotrowski, J., Meinecke, C., Böse, F., 2009. Supporting mobile work processes in logistics with wearable computing. In: Proceedings of the MobileHCI '09. ACM, pp. 1–2.

Schraefel, M., Hughes, G., Mills, H., Smith, G., Frey, J., 2004. Making tea: iterative design through analogy. In: Proceedings of the DIS '04, Cambridge, MA, USA, pp. 49–58.

Sharples, M., Corlett, D., Westmancott, O., 2002. The design and implementation of a mobile learning resource. Personal and Ubiquitous Computing 2002 (6), 220–234.

Simonsen, J., Kensing, F., 1998. Make room for ethnography in design!. Journal of Computer Documentation 22 (1), 20–30.

Skov, M.B., Høegh, R.T., 2006. Supporting information access in a hospital ward by a context-aware mobile electronic patient record. Personal Ubiquitous Computing 10 (4), 205–214.

Smith, M., Cadiz, J.J., Burkhalter, B., 2000. Conversation trees and threaded chats. In: Proceedings of the CSCW'00. ACM, Philadelphia, USA, pp. 97–105.

Smith, M.A., Fiore, A.T., 2001. Visualization components for persistent conversations. In: Proceedings of the CHI 01. ACM, Seattle, USA, pp. 136–143.

Snyder, C., 2003. Paper Prototyping. The Fast and Easy Way to Design and Refine User Interfaces. Morgan Kaufmann, Amsterdam.

Tang, C., Carpendale, S., 2008. Evaluating the deployment of a mobile technology in a hospital ward. In: Proceedings of the CSCW '08. ACM, pp. 205–214.

van den Anker, F.W.G., Lichtveld, R.A., 2000. Early evaluation of new technologies: the case of mobile multimedia communications for emergency medicine. In: Vincent, C., de Mal, B. (Eds.), Safety in Medicine. Elsevier Science, Oxford.

Viller, S., Sommerville, I., 1999. Coherence: an approach to representing ethnographic analyses in systems design. Human–Computer Interaction 14, 9–41.

Viller, S., Sommerville, I., 2000. Ethnographically informed analysis for software engineers. International Journal of Human–Computer Studies 53, 169–196.

Walenstein, A., 2003. Finding boundary objects in SE and HCI: an approach through engineering-oriented design theories. In: Proceedings of the ICSE 2003. Workshop on Bridging the Gap between SE and HCI. IFIP, Portland, USA, pp. 92–99.

Wiltshire, M., 2003. Where SE and HCI meet: a position paper. In: Proceedings of the ICSE 2003, Workshop on Bridging the Gap between SE and HCI. IFIP, Portland, USA, pp. 57–60.

Winograd, T., Flores, F., 1986. Understanding Computers and Cognition. Addison-Wesley.