Group Decision and Negotiation **9**:237–250, 2000

Scenario-Based Design of Cooperative Systems Re-designing an Hospital Information System in Denmark

JAKOB E. BARDRAM
Computer Science Department, Aarhus University, Ny Munkegade 116, Bldg. 540, DK-8000 Århus C., Denmark E-mail: bardram@daimi.au.dk

Abstract

Over the past few years, scenario-based design has attained a growing interest as a way to incorporate a focus on the future use of an application into the construction of software. Scenarios have, however, mostly been used in the design of user-interfaces and hence focused on single-user situations. Based on experiences from applying scenarios in the re-design of an Hospital Information System in the Danish healthcare sector, this paper describes how collaborative scenarios can be used in the design of cooperative computer systems and what such collaborative scenarios should contain. The paper concludes that such scenarios were useful in bridging the gab between understanding collaborative work practices and designing collaborative computer systems.

Key words: scenario-based design, analysis patterns, computer supported cooperative work, Hospital Information Systems, collaborative scenarios

1. Introduction

According to Friedman (1989) the biggest challenge in software development since the 1980s has been to fulfil the needs of the users. According to Winograd (1996) this challenge has in the 1990s been extended to bring *design* to software development in order to ensure that software really *works* – not in the traditional software engineering sense of reliability and efficiency, but in the sense that the *software works for people in a context*. Hence, we would like to work with design requirements for a piece of software that addresses the human activity of using computers for a specific purpose – requirements like easy to learn and use, argument human activities, meet peoples expectations, and become cultural meaningful artifacts. As argued by Carroll (1995) these later requirements are far more difficult to specify and to satisfy. We have little prospect of developing final answers to questions about human activity – and certainly not at the level of detail that would provide specific guidance to designers. Our best course is, therefore, to develop rich and flexible methods and concepts that can incorporate descriptions of users and their current and potential use of a computer system into the very design reasoning about such a system. As a narrative description of what users do and experience when using a computer system, scenarios are such rich description of the human activities that can augment the design of computer systems.

The use of scenarios within Human-Computer Interaction has typically been addressing individual users. Jacobsen's use-cases are, for example, "a sequence of transactions in a system whose task is to yield a result of measurable value to an *individual* actor of the system" (Jacobson 1992, my emphasis). A focus solely on the individual use of a computer system is however too narrow to reveal the conditions for developing and using computer systems, and might even contradict the purpose of contextualizing the design process by applying scenarios. When designing computer support for cooperative work (CSCW) it certainly becomes important to address the collaborative activities at a workplace. The aim of the present paper is to outline such collaborative scenarios. These scenarios were designed and used as a part of redesigning an Hospital Information System in Denmark. The outset for using scenarios in this project originated in the work done in the EuroCoOp and EuroCODE projects at Aarhus University (Bødker et al. 1993; Grønbæk et al., 1995; Kyng, 1995).

2. Scenario-based design

Scenario-based design is useful in situations where the design of the system is fragile in the sense that there is no detailed conception of exactly which work activities should be supported and in which way. Such projects are characterized by high uncertainty and risk, and therefore have to adapt an experimental and iterative way of design (Boehm, 1988). For the purpose of the discussion in this paper there are two central characteristics of such a design process. First, a design process is characterized by *re-designing* existing ways of doing things, which forms the basis for an understanding of how it can be done differently using a computer system. Today, the existing way of doing things already often involves some kind of computer support, which has limitation in its capability of supporting the ever changing work practices. A re-design necessarily has to start by investigating the problems and benefits of the existing system. Second, design is a creative activity that cannot be fully reduced to standard steps. However, a creative process is aided by inspiration, which to a large part comes from looking at the context of future use. Hence, creative design ideas emerge in the *meeting* between the computer professional, drawing on his technological knowledge, and the user, drawing on his or hers knowledge about the work-practices and the organizational setting (see e.g. Bødker and Christiansen, 1994). However, relying on creative ideas to emerge in the juxtaposition of the designers' and users' knowledge in a diffuse high risk design process creates problems of on the one hand to guide the creativity in "the right direction", and on the other hand to decide whether the emerging ideas are so good and creative after all. Hence, in a design process we want to be able to answer questions like; are these design ideas useful, i.e. what kind of work activities do they support and which one do they disturb? How will these design ideas and the system in general fit into the existing organizational context, and how will this context be changed by the system – for good or for worse? How will the system integrate with work practices and instruments that remain unchanged? In these kind of design situations, the benefits of using scenarios are twofold: on the one hand they are vehicles for supporting the creative meeting between designers and users, and on the other hand they help answer the questions on the usefulness of a system compared to the work practices within an

organization. Let us consider what scenarios aimed at describing collaborative work activities should entail.

2.1. Collaborative scenarios

The purpose of collaborative scenarios is to provide support for the overall design of a computer system by describing collaborative work activities that are to be supported and/or affected by the future computer system. Such scenarios are work-driven, open-ended and informal narratives of what people do and experience as they try to perform different activities with or without making use of a computer application. Despite their popularity, there is no general accepted definition of what a scenario is, what it should entail, or how it should be used – even the inclusion of the "computer" in scenarios is not always taken for granted (see e.g. the definition in Karat (1995)). However, the definition provided by Carroll (1995) makes a good starting point for discussing what a collaborative scenario should cover:

> "The defining property of a scenario is that it projects a concrete description of activities that the user engages in when performing a specific task, a description sufficiently detailed so that design implications can be inferred and reasoned about. Using scenarios in system development helps keep the future use of the envisioned system in view as the system is designed and implemented; it makes use concrete – which makes it easier to discuss use and to design use." (p. 4).

This broad definition of a scenario however raises both interesting and difficult questions: First, system development has always been guided by descriptions of potential new ways of supporting and enhancing work by computer technology. So what is new in using scenarios and how does they differ from traditional requirement specifications? Second, what is meant by "a concrete description of activities"? What is meant by an activity? How concrete should the description be? What should this description contain? Third, what is important to write down in a scenario so that "design implications can be inferred and reasoned about"? What kind of implications are we talking about? The kind of implications that we would like the system to have or the unwanted kind of implications that just seems to come anyway? What is the role of the computer system in the scenarios describing activities? Finally, how can a scenario, as a narrative description on a piece of paper, "envision" a future use situation that is not even quite envisioned by the designer, let alone the user? And what is meant by "discuss use" and "design use" – with whom should we discuss and design use?

Now these are general and far-reaching questions and the scope of this paper clearly do not allow a detailed discussion of all of them. Therefore, I shall concentrate on the second and third question and shortly comment on the last one. Answers to the first question has been discussed extensively by the different authors in the book "*Scenario Based Design*" edited by Carroll (1995) and in different paper in the Journal on Human-Computer Interaction (e.g. Bürkle et al. 1995).

3. The SAIK Project: Computer support for coordinating medical work

The SAIK project[1] was launched as an experimental pilot project at Kommunedata in an attempt to redesign a national-wide Hospital Information System called the Green System (GS). Currently GS is a large mainframe-based information system used by most hospitals in Denmark. The aim was to redesign GS into a client-server architecture, preserving the mainframe technology as a database server but building PC-based client applications dedicated to support the work at different departments within an hospital – e.g. at the emergency and casualty department, at a medical ward, and at a surgical planning office. One of the main problems within Danish hospitals today is coordinating the treatment made in the different departments. The purpose of the SAIK project was to investigate how coordination and planning of patient care happens today – both with and without computer support – and based on these investigations to reveal how this coordination can be supported by computer technology. The Patient Scheduler is a prototype that illustrates how the coordination of healthcare work can be coordinated by computers.

3.1. Methods and scope of the investigations

The SAIK project took place over a period of two years, involving 5 different hospitals in Denmark. The project had two main strands: ethnographic inspired workplace studies of the cooperative nature of work within hospitals, and a participatory design process developing the Patient Scheduler. The workplace studies and preliminary data analysis were based on qualitative methods such as qualitative interviews; participative observations of daily work at the ward, meetings and conferences (cf. Patton, 1990); and studies of different documents, records and other tools (Jordan, 1996). Field studies were made in 5 hospitals. Two of these hospitals were incorporated in the participatory design process of the Patient Scheduler applying methods such as future workshops (Kensing and Madsen, 1991), cooperative prototyping session (Bødker and Grönbæk, 1991) and organizational prototyping (Bardram, 1996).

4. Design documentation

Now let us turn to a description of the different design documents used to sustain the experiences obtained during all these activities. Figure 3 contains a summary of the documents. Please note that when using the word “document” we do not solely mean written documents; documentation in the form of photographs, video, rich pictures, process flow-charts, and photocopies of different paper-based forms, documents, work protocols, etc. were central parts of the documentation. The documentation can be divided into two broad categories: (i) organizational overviews, providing a description of the organizational context of the collaborative work-processes; and (ii) work activity scenarios, which are scenarios trying to capture the collaborative work, which we are designing for.

4.1. Organizational overviews

An insight into the organization where the future computer system has to be implemented and where the system development process has to take place is clearly indispensable. Thus several authors stress the need for "getting to know the domain, people and tasks" (Johnson et al., 1995 p. 214) and the need for general "work situation overviews" (Kyng, 1995). These kinds of descriptions are essential because a specific work-task scenario is only given meaning from the situation in which it is used. In our project we used four different kind of representation of the hospitals as organizations:

Organizational overview (OO). The OO is intended to provide a sufficiently detailed description of the tasks, goals, purposes, and strategies of the organization, the types of jobs and the roles within the organization, how the employee are organized (structure), and the different kind of technology used there. The environment of the organization in terms of competitors, society, labor unions, etc. is part of such an organizational overview as well as necessary descriptions on cultural systems of status, prestige, etc.

Person-oriented record (POR). The POR is intended to capture the work practices of a person – both a particular person and a generic job description: the sequence of actions and tasks in the daily round, who they collaborate with, their responsibilities and job descriptions, what they perceived as routine and exceptional work, how they handle exceptions and problems, and so on.

Object-oriented record (OOR). The OOR is intended to describe the construction and career of an object, artifact, or document through the system: how is the object created, what does it consists of, what locations does it "visit", who owns it, what other objects is it depending on and in contact with, who has the right to manipulate it, change it, remove it, and so on.

Setting-oriented record (SOR). The SOR chronicles what happens in a particular location through time – throughout a shift, a day, a week, or other relevant temporal cycle to the workplace in question. Many kinds of work activities are spatially distributed and the SOR is intended to capture the work taking place in these separate locations.

4.2. Work activity scenarios

In a system development project the organizational overviews are typically made once and for all. In contrast hereto the Work Activity Scenarios (WAS) are alive during the whole system development process. They are constantly modified and rewritten according to new understandings of the work practice and according to the evolving design of the computer system. Hence, we maintain two sets of activity scenarios: one set of scenarios of current work activities and one set of scenarios of the envisioned future work activities. This might sound as a lot, but often the introduction of a computer system might not

A detailed analysis of *one* collaborative work activity entails asking:

- ***what is done and why*** – analyzing the product and purpose of the activity from the point of view of the organization and the from the point of view of the different groups of involved actors
- ***what sub-actions*** are part of the activity – analyzing the detailed flow of distributed tasks of all participants
- ***how*** are these sub-actions realized using different tools and artifacts – analyzing the role of artifacts and their mediating and coordinating role, and how these artifacts interrelate
- ***who*** are responsible for these sub-actions – analyzing the division of work
- ***where*** and ***when*** are these sub-actions done – analyzing the spatial and temporal arrangement of the activity, crossing departmental, organizational or geographical boundaries to encompass all the actors involved

A detailed analysis of *several interdependent* collaborative activities involves asking:

- how is the routinely flow of work continuously coordinated in terms of the three basic types of coordination: (i) *communicative coordination*, where the actors coordinate through signs and language, (ii) *instrumental coordination*, where each actors coordinate his activity according to the activities of others, and (iii) *scripted coordination*, where each actor coordinates his activity according to a script for action, e.g. a checklist, a plan or a schedule.
- what kind of *collaborative breakdowns* happens in the daily flow of work, looking at:
  - when and how the routinely *coordinated* work collapses caused by necessary accommodation to unforeseen constraints in the working situation and how the coordinated flow of work is reestablished through a mutual cooperative effort.
  - when and how *cooperative* work breaks down caused by conflicting motives and goals and how this situation is handled through rethinking and co-constructing of the activity system.
- how are the different activities *interdependent* in terms of the three general types of interdependencies: the need for (i) simultaneous activities, (ii) sequential activities, and (iii) shared resources in activities.
- what are the contradictions and conflict within and between existing work activities and between existing work activities and potential new ones supported by the computer technology

Figure 1. A checklist for creating collaborative scenarios.

change much in the overall activity system, and if it does, such changes has to be considered and described anyway. WASs are scenarios that detail the activities necessary to get a particular task or process within the total scope of work done. The WAS has a unique name in order to facilitate communication among designers, users, stakeholders, etc. A WAS describe the recurring, regular features of typical tasks and how they relate to the organizational context and to the physical setting, facilities and persons at the workplace. The scenarios are non-technical and encompass both individual as well as collaborative work tasks. They have the purpose of requirement analysis, environment for the overall design decisions, and provide the basis for all further scenarios. The future WASs are also used for implementation and training.

A WAS is produced by workplace studies and participatory design techniques. Figure 1 shows a checklist of aspects of collaborative work that a collaborative work scenario should

address. This checklist has been compiled from Activity Theory as a framework for design of CSCW systems (see Bardram, 1998), and the insights from our workplace studies and from numerous other workplace studies done within CSCW (for an overview of some of the findings see Plowman et al., (1995) and Grinter (1997)).

Some work activities are central to the (re-)design of a computer system and hence need to be analyzed in greater detail. For this purpose *Analytical Scenarios* (AS) can be made. An example of an analytical scenario is illustrated in figure 2. The analytical scenario describes in detail what is happening, where and when, by who and why, and how both today and potentially in the future supported by a computer system. Relevant information from the organizational overviews are included (e.g. the description of the responsibility of the head radiologist) and the underlining are references to other description (e.g. the SOR describing the offices). The last column (the "**How – Patient Scheduler**") is added later when the design is evolving and illustrated partly by prototypes or mock-ups. Even if the design obliterates some subtasks, these are kept in the analytical scenario as a reminder of how the future system will change and potentially enhance work. The sentences in italic are used to comments for further action in the design of the computer system – e.g. there might be a serious problem in not supporting the central task of prioritizing incoming requisitions. In this case we have a *contradiction* between the current and the future scenario as supported with the current design of the prototype.

4.3. Activity maps

As a way of providing an overview of all interdependent and contradicting activities activity maps were drawn. These maps were merely a graph with activities as nodes and relations in term of interdependencies and/or contradiction as arcs. Activity maps were drawn both of the current work situation and of the envisioned future work situation.

5. Applying scenarios in the SAIK software design process

In the SAIK project an evolving set of scenarios constituted a backbone, tying together the many activities in the system development lifecycle. This is in contrast to authors advocating the use of scenarios only for workplace descriptions and initial requirement specification (Anderson and Durney, 1992; Hsia et al., 1994) or for evaluation (Nielsen, 1995). Central to our approach is to use scenarios for describing existing work situations and then use these to help generate a system solution and for continuous verification of the design through experimentation. In the SAIK project collaborative scenarios played three overall purposes: (i) continuous analysis and design documentation, (ii) validation of design solutions and experimentation with prototypes, and (iii) generalization of experiences in order to re-use design insights and solutions in other design projects.

5.1 Scenarios as the fulcrum in the design process

In the SAIK project we operated with a design process consisting of three activities: (i) exploration, (ii) design, and (iii) experimentation. We alternated between these activities in an iterative way, trying to use the experience obtained in one activity as an input for the other activities.

In the *exploration* activity the necessary insight into the overall socio-economical and organizational context was initiated. Understanding the Danish hospital sector and its political and economical nature was of central importance to the SAIK project and this exploration was hence never terminated, but continued throughout the whole project. This organizational insight was documented in the organizational overviews.

The work activity scenarios of existing ways of doing work were also created in this activity. In the SAIK project central work processes for collaboration and communication at the hospital was described. Because the Patient Scheduler was aiming at supporting the cooperation across departmental boundaries, we wrote scenarios concerning the requisition of radiology examinations, the collaboration among physicians at different conferences, the planning and booking of examinations at the radiology department, etc. Subsequently, analytical scenarios were made for these central work processes (see e.g. figure 2). However, scenarios "at the border" of such central activities for the Patient Scheduler were made as well, e.g. the way medication was given at the ward, and how the physician was using the medical record. The organizational overviews and the work activity scenarios were compiled into the activity maps. These maps were in practice the walls of our offices. Scenarios, pictures, screen-dumps, and description of different artifacts (mostly documents and forms) used at the workplace were put on large bulletin boards, and the connections between all this were maintained by red yarn and post-its notes.

At some point concurrent with the exploration activity, the *design* activity is initiated. This is a creative process of generating ideas for computer support, which is guided by the obtained insight in the exploration activity. Design decisions are facilitated by the different work activity scenarios, which point to issues in the current way of working that need to be considered. For example, the scenario describing the activity of scheduling examinations at the radiology department shown in figure 2 pointed to the need of supporting the sorting of incoming requisitions. This design decision was subsequently evolving into support for setting up some kind of automatic filtering according to sender, type of request, etc. For each of the work processes, that we were trying to support, future scenarios were used to document how the computer system might enhance, change, or obliterate existing work activities. These future scenarios are changed, up-dated and used throughout the whole design and construction phase of the computer system. For example, in the SAIK-project it was decided that the Patient Scheduler should support "both ends of the collaboration" – i.e. that it should support both receiving and sending requests for work at other departments. Hence, future scenarios for both the work at wards and at radiology and other service departments were written. Furthermore, a design solution allowing the physician at the ward to book examinations at radiology on his own was made. This would save both the physician and the secretaries at radiology a lot of work. However, this was a

What	Where & When	Who & Why	How – Today	How – PATIENT SCHEDULER
Receiving paper requisitions	In the office of the radiology department. All day		The ward requesting the examination places it into an inbox at the counter at the office. There are 4 kind of inboxes ...	Paper-based requisitions have to be entered manually into the system. *NOTE: We need to consider connection to EDIFACT messages!*
Printing out electronic requisitions	Using the Green System at the main desk in the office.	A paper copy is printed because: (i) the head radiologist need a paper copy, (ii) it needs to be present during the examination, (iii) it has to be achieved in the big filing cabinet.	Entering the 'Receiving requisitions' form in the Green System. Typing 'pr' outside each req. that should be printed. ...	Select (multiple select) requisition in folder. Press Menu|Print or Press Crtl+P or press the print icon in the tool bar.
Sorting requisitions			The secretaries department sorts the requisitions according to several criteria, like what kind of examination ordered, levels of urgency, the type of illness of the patient, etc.	Sort the list of requisitions according to sender, type of examinations, etc. Creating new folders in a hierarchy of folders. Sharing folders among users.
Prioritizing examinations	At the head radiologist's office.	The head radiologist, the head nurse, and a secretary The priority, and thereby the order that patients are being examined are solely the responsibility of the head radiologist.	ÅKH: the secretary has made a pre-priority because she knows how typical patients are prioritized by the head radiologist. This saves some time, because the head radiologist only has to concentrate on the special cases.	To each requisition a dispersal note can be attached telling how this case should be handled. *NOTE: We cannot explicitly support the prioritizing of requisitions!*

Figure 2. An example of an analytical scenario.

radical new solution to the communication between wards and radiology, and several future scenarios were made to envision how this would be possible.

5.2. Design experimentation and confrontation

A central part of an iterative design process is to make *experiments* in order to clarify the overall design of the system and to investigates the qualitative aspect of usability, acceptability, and suitability within the target domain(s). For this purpose we operated with four kind of *design confrontations*: (i) validation, (ii) logical confrontation, (iii) use confrontation, and (iv) organizational confrontation.

Validation is a confrontation between the understanding obtained during the exploration activity as documented in the organizational and work-oriented descriptions and scenarios.

In other words, it is a validation of the correctness of the obtained descriptions by discussing the scenarios with the users. This validation is of crucial importance when the scenarios are to be used for further development and design. In the SAIK project this validation was achieved by reviews of documents and video analysis, and in workshops with different employees who has participated in the exploration phase.

A *logical confrontation* happens between a proposed design and the analytical scenarios. The confrontation aims at pinpointing the potential opportunities and risks of the future system according to the way work is done today. The confrontation is called logical because it is a systematic comparison of a proposed design with the knowledge about the work practice of today. Two examples of logical confrontations are illustrated in the analytical scenario in figure 2 (shown in italic). These confrontations reveal problems of connecting the Patient Scheduler to EDIFACT messages coming from outside the hospital, and problems of supporting the prioritizing of requisitions. Thus, these confrontations pointed to potential risks in the overall design.

A *use confrontation* happens between a proposed design as documented by the future scenarios and the future users of the computer system. In the SAIK project we enacted the future scenarios together with the users, using different prototypes illustrating the future system. This confrontation aims at revealing the use-characteristics of the system, potential problems and opportunities for further design. The future scenarios were changed and enriched together with suggested changes to the prototypes.

An *organizational confrontation* happens between a proposed design – either illustrated by a prototype or by the final system – and the organizational context of the new system. The aim is to reveal how the computer system supports, enhances and changes the working of the organization as a whole. Thus the system has to be evaluated and confronted with work practices, the organizational structure and culture, related technology, resource constrains, spatial arrangements in the workplace, etc. In the SAIK-project this kind of confrontation was made in different workshops with managerial representatives. In one of these workshops the design solution of having physicians directly book radiology examinations on their own was discussed and found highly problematic from the radiology departments point of view. Several problems with the solution were revealed, ranging from the fact that a physician cannot book all radiology examinations without advice from a radiologist, to more economical issues of how the radiology department can control their expenses if everybody were granted access to book on their own. Hence, there was a need for designing a solution for keeping radiology in control. This involved creating an access mechanism, so that radiology could decide exactly what kind of examinations could be booked by who, when, how, etc.

5.3. Analysis patterns – generalizing design knowledge

In the SAIK project, the analytical scenario has two distinct purposes in the system development lifecycle: (a) as detailed task analysis of work practices of central importance to the design, and (b) as generalizations of experiences from the different hospitals involved in the project. The last purpose must be viewed in light of the overall aim of the SAIK

project to design a system that not only should be used at the hospitals involved in the design process, but potentially at all hospitals in Denmark. When deviations were discovered they were kept in the scenario – for example the sentence in figure 2 marked with "**ÅKH**:" is an observation made only at this hospital. The scenarios made during our investigations helped us on the one hand to identify and sustain the differences in work practices across different departments and hospitals. On the other hand the scenarios captured aspects of collaborative work that were stable and similar across different work settings, and they could be generalized into generic scenarios for different types of activities within an hospital. Examples of such generic types of activities are the paper-based requisition of radiology examinations and the scheduling and sorting of incoming requisitions. These generic scenarios provided the background for extracting the general design knowledge embedded in the Patient Scheduler as *Analysis Patterns*, which could be reused in other projects at Kommunedata. In contrast to Design Patterns (Gamma et al. 1995), an analysis pattern is a solution to a recurrent problem within an *organizational* context, not within construction of software. An analysis pattern is an object-oriented solution that represents a common construction in some business modeling – in this case within hospitals as an organization (see also Fowler, 1997). The real-world problem, that each analysis pattern is attempting to solve, is represented as a generic scenario, which work as an inspiration for the analyst in the future.

6. Conclusions

The collaborative scenarios, as discussed in this paper, are summarized in figure 3. In conclusion, using collaborative scenarios as the backbone in the design of a cooperative system in the SAIK-project, has been very successful. They provided a necessary tool for analyzing and documenting existing work practices and hence paved the way for generating ideas for new or redesigned computer support for these work practices. But most important, collaborative scenarios worked as important thinking tools for grounding the creative envisioning of how work could be organized using new computer technology. As such imaginary thinking tools helping to answer the question of "given this design proposal, what might be the future use of the system?", collaborative scenarios were a fundamental cornerstone in the participatory design sessions with the users (see Bardram, 1996). Moreover, scenarios are not "dead" documentation, but are alive throughout the whole design process and provides the basis for later construction of software and the final implementation of computer systems within organizations. In this way, scenarios can mediate an implementation and diffusion process of computer technology within an organization, by translating existing work practices into new ones using the system. Note that an implementation phase influences the creation of the organizational and work-oriented overviews (see figure 3). This is a result of the spiral model where experiences obtained during the phase of implementation – as the process of turning a computer system into technology for an organization – might provide a basis for further redesign or implementation of the system, either in the same organization or in similar organizations (e.g. another hospital).

Design Documents	Description	Produced in design activity	Special attention to
Organizational Overview			
Organizational Overview	Provides the organizational and domain description necessary to understand the setting in which the system to be developed and later used.	Exploration (& Implementation)	Organizational, Economical, Socio-political, Cultural Context of the collaborative work activities.
Person-oriented records	Records the work practice of a person or a type of job within the organization	Exploration (& Implementation)	Division of work, Roles, Status. Responsibility, Location, Managerial Positions.
Object-oriented records	Describes an artifact within the work setting, what it is made up of, who does what with it and when, etc.	Exploration (& Implementation)	Which tools are used for which activities, sharing of tools, coordination through tools, the history of tools
Setting-oriented records	Describes a particular setting within the organization and its relation to other locales.	Exploration (& Implementation)	Physical arrangement of setting, the location of things, such as desks, telephones, wallboards, etc.
Work Activity Scenarios			
Activity Map	Describes the overall web of activities, their interrelationships, and the relationship to the organizational context. Is used to interlink the organizational and work-oriented overview to the activity scenarios.	Exploration, Experimentation (& Implementation)	The interdependencies among activities, the contradictions between activities, and the relationships between existing and future computer supported activities
Work Activity Scenarios	Provides an overview of the outcome of an activity, the purpose of it, how it is accomplished.	Experimentation & Construction	The purpose and outcome of the activity, the constraints put on the activity.
Analytical Scenarios:	Provides a detailed analysis of a collaborative activity including its division into different subactivities and how these are distributed onto several actors, using different artifacts, are located in time and space, and each having a specific purpose.	Experimentation & Construction	The division of work in a particular activity, the detailed use of artifacts, the distribution in terms of actors, time and space, inherent problems and contradictions in the activity.
Analysis Pattern	Provides a generic solution to a specific design problem within a domain. Contains a scenario, describing the problem setting, and a OO diagram and a user-interface describing the design solution	Experimentation & Construction (& later reuse)	Recurrent pattern in scenarios. Design solutions that are validated as useful through experimentation.

Figure 3. Design documentation summary.

Another, more theoretical, conclusion to be drawn from our use of scenarios is that they provide one solution to bridge "the great divide" within CSCW. This term labels the division between CS (Computer Support) and CW (Cooperative Work), the former focusing on technical innovations, and the later on social aspects of work. The problem with this division is that neither of the sides have been focusing on the process of getting from the one to the

other – i.e. have not been addressing neither the issue of *designing* computer systems based on understanding cooperative work, nor the issue of *implementing* computer systems within cooperative work practices. Within the numerous workplace studies made within CSCW it is often argued that one of the main strengths of an ethnographic approach is that detailed analyses of social work can provide rich material on which to base recommendations for the design or re-design of a computer system. However, there is a big distance from having a good understanding of existing work practices to creating design solution for a future computer system, which is intended to *change* these work practices. Typical design recommendations from such ethnographic workplace studies is enclosed as the classic "implications for design" section at the end of the paper (c.f. also Plowman et al., 1995). In the SAIK-project collaborative scenarios proved to be a good way of both documenting experiences obtained during the workplace studies and at the same time they worked as design tools, helping to bridge the distance between present and future work practices. Moreover, as already emphasized, scenarios are "live documents" that are used active in cooperation with users. In this way, scenarios support a two-way communication between designers and users, where users inform designers about current work practices and designers inform users of potential future computer solutions. Hence, design and implementation is facilitated concurrently. Such a two-way communication process is fundamental distinct from the classical use of workplace studies within CSCW, where the ethnographers are the ones in contact with the real work setting, and they are informing the design process through "debriefing meetings" (Hughes et al., 1994).

Acknowledgments

The work done in the SAIK-project is funded by the Danish Academy of Technical Sciences, Kommunedata, and the University Hospital for Århus Amt. Part of the workplace studies, the scenarios, and the workshops were made in cooperation with Trine Grundahl, Jens Due Olsen, and Marian Thygesen. We are grateful to all personnel at the 5 hospitals for their willingness to have us hang around asking obvious questions and for their participation in the design process.

Notes

1. SAIK is a Danish abbreviation for "Collaborative Informatics in Clinical Practice."

References

Anderson, J. S., and B. Durney. (1992). Using scenarios in deficiency-driven requirements engineering. *Proceedings of the IEEE International Symposium on Requirement Engineering*. Washington: IEEE Computer Society Press, 134–141.

Bardram, J. E. (1996). Organisational Prototyping: Adopting CSCW Applications in Organisations. *Scandinavian Journal of Information Systems*, 8, 69–88.

Bardram, J. E. (1998). *Collaboration, Coordination, and Computer Support: An Activity Theoretical Approach to the Design of Computer Support for Cooperative Work*, Doctoral Dissertation, University of Aarhus.Available as Daimi Technical Report, PB–533.

Boehm, B. W. (1988). A Spiral Model of Software Development and Enactment, *IEEE Computer*, May 1988, pp. 61–72.

Bürkle, U., G. Gryczan, and H. Züllighoven. (1995). Object-Oriented System Development in a Banking Project: Methodology, Experience, and Conclusions. *Human-Computer Interaction* 10: pp. 293–336.

Bødker, S., and K. Grønbæk. (1991). Design in Action: From Prototyping by Demonstration to Cooperative Prototyping. In J. Greenbaum and M. Kyng (eds.), *Design at Work: Cooperative Design of Computer Systems*. Hillsdale, New Jersey: LEA.

Bødker, S., E. Christiansen, K. Grønbæk, K. Halskov Madsen, P. Mogensen, M. Robinson, H. Kühn, S. Robinson, M. Türing, E. Hinrichs, P. Sørgaard, and P. Hennessy. (1993). *EuroCODE D-1.1: The EuroCODE Conceptual Framework*. Aarhus University, 17 June 1993. CODE-EMP-93-1.

Bødker, S., and E. Christiansen. (1994). Scenarios as Springboards in Design CSCW, *DAIMI PB* no. 488, University of Aarhus.

Carroll, J. M., ed. (1995). *Scenario Based Design: Envisioning work and technology in system development*. New York: John Wiley and Sons, Inc.

Fowler, M. (1997). *Analysis Patterns: Reusable Object Models*. Reading, MA: Addison-Wesley.

Friedman, A. (1989). *Computer Systems Development: History, Organisation and Implementation*. Chichester: Wiley.

Gamma, E., R. Helm, R. Johnson, and J. Vlissides. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, MA: Addison-Wesley.

Grinter, R. (1997). From Workplace to Development: What Have We Learned So Far and Where Do We Go?, In Proceedings of ACM GROUP'97 Conference, Phoeniz, AZ, ACM Press.

Grønbæk, K., M. Kyng, and P. Mogensen. (1995). Cooperative Experimental System Development -Cooperative Techniques Beyond Initial Design and Analysis. *Proceedings of Computers In Context, Aarhus, Denmark*. Department of Computer Science, Aarhus University.

Hsia, P., J. Samuel, J. Gao, D. Kung, Y. Toyoshima, and C. Chen. (1994). Formal approaches to scenario analysis. IEEE Software (March); p. 33–41.

Hughes, J., V. King, T. Rodden, and H. Andersen. 1994. Moving Out from the Control Room: Ethnography in System Design. *Proceedings of CSCW 1994, Chapel Hill, North Carolina*. ACM.

Jacobson, I., M. Christersson, P. Jonsson, and G. Övergaard. (1992). *Object-Oriented Software Engineering – A Use-case Driven Approach*. Reading, MA: Addison-Wesley.

Johnson, P., H. Johnson, and S. Wilson. (1995). Rapid Prototyping of User Interfaces Driven by Task Models. In John M. Carroll, (ed.), *Scenario Based Design: Envisioning work and technology in system development*. New York: John Wiley and Sons, Inc.

Jordan, B. (1996). Ethnographic Workplace Studies and CSCW. In D. Shapiro, M. Tauber, and R. Traunmüller, (eds.), *The Design of Computer Supported Cooperative Work and Groupware Systems*. Amsterdam: Elsevier.

Karat, J. (1995). Scenarios Use in the Design of a Speech Recognition System. In J. Carroll (ed.), *Scenario Based Design: Envisioning work and technology in system development*. New York: John Wiley and Sons, Inc.

Kensing, F., and K. Halskov Madsen. (1991). Generating Visions: Future Workshops and Metaphorical Design. In J. Greenbaum and M. Kyng (eds.), *Design at Work:Cooperative Design of Computer Systems*. Hillsdale, New Jersey: LEA.

Kyng, M. (1995). Creating Contexts for Design. In J. Carroll, (ed.), *Scenario Based Design: Envisioning work and technology in system development*. New York: John Wiley and Sons, Inc.

Nielsen, J. (1995). Scenarios in Discount Usability Engineering. In J. Carroll (ed.), *Scenario Based Design: Envisioning work and technology in system development*. New York: John Wiley and Sons, Inc.

Patton, M. Q. (1990). *Qualitative Evaluation and Research Methods*, 2nd edition. Newbury Park: SAGE.

Plowman, L., Y. Rogers, and M. Ramage. (1995). What Are Workplace Studies For? *Fourth European Conference on CSCW, Stockholm*, Kluwer Academic Publishers.

Winograd, T. ed. (1996). *Bringing Design to Software*, New York, NY: ACM Press.