

The Use of Cooperation Scenarios in the Design and Evaluation of a CSCW System

Oliver Stiemerling and Armin B. Cremers

Abstract—Design and evaluation of groupware systems raise questions which do not have to be addressed in the context of single user systems. The designer has to take into account not only the interaction of a single user with the computer, but also the computer-supported interaction of several users with each other. In this article we describe the use of cooperation scenarios in the design and evaluation of an innovative access control system for a concrete groupware application developed in the POLITeam project. We have used informal textual scenarios to capture a rich description of the particularities of access to cooperatively used documents in three different organizations. Based on these scenarios, we have developed an access control system, which not only allows specification of access rights in advance but also allows involvement of third persons at the actual time of access, using negotiation and notification mechanisms. We describe our evaluation strategy which again employs the cooperation scenarios developed in the empirical phase. After relating our approach to other work, we summarize and discuss our experiences and the advantages (and disadvantages) of using scenarios for the design and evaluation of Computer Systems Cooperative Work (CSCW) systems. Finally, we give a brief outlook on future work.

Index Terms—Access control, cooperation scenarios, CSCW, design methodology, evaluation, groupware, scenario-based design.

1 INTRODUCTION

COMPUTER SYSTEMS COOPERATIVE WORK (CSCW) is an interdisciplinary field of research, dealing with cooperative work supported by computer systems. (For an overview see e.g., [6].) Computer science naturally is a major contributor. However, psychology, sociology, and other disciplines are involved, as well. The developers of CSCW systems (groupware) not only have to design for single users interacting with the system but also for groups interacting *via* the system. This collaborative perspective raises or amplifies, for instance, issues like *privacy and access control* ([9], [23]), *conflicts* ([27]), *awareness* ([4], [7]), and *tai-lorability* ([21], [22]), which we discuss briefly in the following in order to give the reader an overview of some of the challenges in groupware design.

As early as 1986, in the context of the first CSCW conference, Greif and Sarin ([9]) stated that the access control mechanisms and concepts used in operating systems of that time were not flexible enough to express access policies for group interaction. They suggested the development of more sophisticated controls taking into account additional factors like user roles, access rights concerning abstract operations other than read and write (e.g., sharing operations), and specific object-user-relationships (e.g., the current or past users of an object). Shen and Dewan ([23]) discuss the problem of access rights for collaborative work in the context of the multiuser editor SUITE. They stress the dynamic nature of collaborative work and thus state the support of

multiple, dynamic user roles and the need for easy specification of access rights as important requirements for access control in groupware. Furthermore, they introduce specific collaboration rights for operations whose results can affect other users.

The fact that groupware functionality can affect multiple users gives rise to a strong potential for conflict concerning the configuration and use of such systems ([27]). Groupware systems can distinctively change the division of labor in an organization, obliterate jobs, and open new opportunities for communication between employees (a development which might not be in the interest of management). Furthermore, supporting communication and coordination with computers opens new ways of controlling and monitoring work. Often small things like the publically visible creation date of a document are cause for vehement negative reactions of groupware users.

While groupware systems allow for more control over work processes on one hand, on the other hand important context information can be obliterated by cooperating e.g., via a shared document work space. Even small clues like seeing the overflowing (physical) inbox of a colleague or hearing the voice of a somebody in the hallway might have formerly been helpful in deciding questions like whom to send which document or when to ask for the completion of some piece of work. When supporting cooperative work with computers, this raises the issue of *awareness* (see e.g., [4]). Groupware users sometimes need to be aware of what is going on in the systems and what other users are currently doing or have done in the past. Designing mechanisms to provide awareness in groupware is nontrivial because one has to walk a tight rope concerning the conflicting goals of privacy vs. the need for awareness and information overflow vs. lack of awareness. Fuchs et al. ([7]) present as one solution to this problem a model for an

• O. Stiemerling and A.B. Cremers are with the Department of Computer Science III, University of Bonn, Roemerstrasse 164, 53117 Bonn, Germany. E-mail: {os, abc}@cs.uni-bonn.de.

Manuscript received 18 Dec. 1997; revised 7 July 1998.

Recommended for acceptance by R. Kurki-Suonio and M. Jarke.

For information on obtaining reprints of this article, please send e-mail to: tse@computer.org, and reference IEEECS Log Number 107450.

awareness mechanism which is characterized by a high degree of tailorability, i.e., it can be adapted to different and dynamically changing needs and preferences of individuals, groups, and organizations.

While tailorability (or adaptability) is also an issue outside CSCW, the complexity, dynamics and diversity of cooperative work increase its importance in this field beyond the configuration of awareness mechanisms (see e.g., [21], [22]). Furthermore, the fact that groupware functionality has the potential to affect multiple users raises the question of who is allowed to tailor this functionality and how one can explore and try out tailored functionality without disturbing other users.

Access control, potential for conflicts, awareness, and tailorability are some examples (and by no means all) of the issues which arise in the design of CSCW functionality. However, when extending the use of computers from the support of single users to the support of groups, not only the functionality but also the development processes have to be rethought. Grudin [10] gives an extensive overview of the novel challenges groupware designers have to face. Among others, he identifies the need for almost unanimous acceptance of groupware in order to achieve a critical mass of users and a well balanced benefit profile (e.g., subordinates gain as much from the system as from managers) as key factors to the success of groupware. Thus, groupware developers have to know much more about the context of use of the prospective application than developers of single user applications, who do not have to pay as much attention to group-related aspects like, for instance, trust, awareness of each other, negotiations, social dynamics, power structures, and work processes. Trust, awareness, and negotiation are aspects which proved highly relevant in the design case described later on.

Grudin [10] also points out the difficulties in evaluating groupware. The interaction of a single user with a system is much easier to evaluate in a laboratory setting than group processes. Apart from the sheer logistical problems of getting even a small-sized group in the same lab at the same time or the nightmare of installing a prototypical application in an organization, the necessary duration of the evaluation is a major problem, because “*group interactions unfold over days or weeks*” ([10, p. 101]). In the following, we want to concentrate on the two design process related problems identified above: capturing of rich contextual requirements and evaluation support.

In the POLITeam project (see [17]), we employ textual scenarios drawn from field studies, interviews, and workshops to inform the design processes and support the evaluation prior to field tests and large scale workshops. We call our scenarios *cooperation scenarios*, as they not only capture the work and interaction of single users with the system but also the group and organizational context and its work practices.

Within the POLITeam project a groupware application for a German federal ministry and selected ministries of a state government and the concurrent engineering division of a car producer is developed in an evolutionary and participative way. The first system version was generated by configuring the commercial product LINKWORKS by Digital.

Based on the experiences gained by introducing the first system version in three different fields of application, we developed advanced versions of the system. The functionality mainly consists of an electronic circulation folder, shared workspaces, and an event notification service.

In this article, we describe the use of cooperation scenarios in the design of the access control system in POLITeam. Specifically, we want to point out the value of cooperation scenarios for the creation of novel CSCW-functionality, in this case the integration of a traditional anticipative access control system with *negotiation* (computer mediated decision making, see also [27]) and *notification* services. In section two we describe the concept of cooperation scenarios in more detail, comparing it with other types of scenarios in the related literature. Section 3 introduces the access control design problem and presents three cooperation scenarios which have notably guided and motivated the new design. Section 4 contains the analysis of the scenarios and briefly outlines the resulting implementation. Section 5 describes the use of scenarios in three consecutive evaluation steps. Section 6 relates our approach to other work. Section 7 summarizes and discusses the value of cooperation scenarios. Finally, Section 8 suggests future research efforts.

2 COOPERATION SCENARIOS FOR THE DESIGN AND EVALUATION OF CSCW SYSTEMS

Carroll [3] identifies several different roles which scenarios can play in the development process of software. We employ scenarios in the CSCW design process in three roles: as a tool for the first (informal) requirements analysis, for communication support (user-designer and designer-designer) during validation, and for evaluation support. Fig. 1 shows how the scenario supported steps are positioned in the overall design process. Note that the process is cyclical and that we believe the value of cooperation scenarios lies foremost in the first cycles when attempting to identify or create innovative functionality which supports cooperative work. Later cycles rely on more formal methods and models.

2.1 The Form of Cooperation Scenarios

Scenarios can take many different forms. Kuutti [18] describes two extreme positions concerning the scope of scenarios. The first one “*sees a scenario as an external description of what a system does*” (p. 21), while the second one looks “*at the use process as situated in a larger context*” (same page). According to these different positions, scenarios can take the form of exact protocols or formal sequence descriptions on one hand, and rather broad, mostly textual descriptions also covering contextual aspects which are only loosely related (however, relevant) to system design on the other hand (also see [3]).

For our purpose, we need a form of scenario which allows designers to capture a broad range of (eventually) unanticipated, contextual information. Thus, our cooperation scenarios are based on informal textual descriptions of work practices, including the motivation and goals behind cooperation. The textual form also facilitates the discussion between users and designers (role of com-

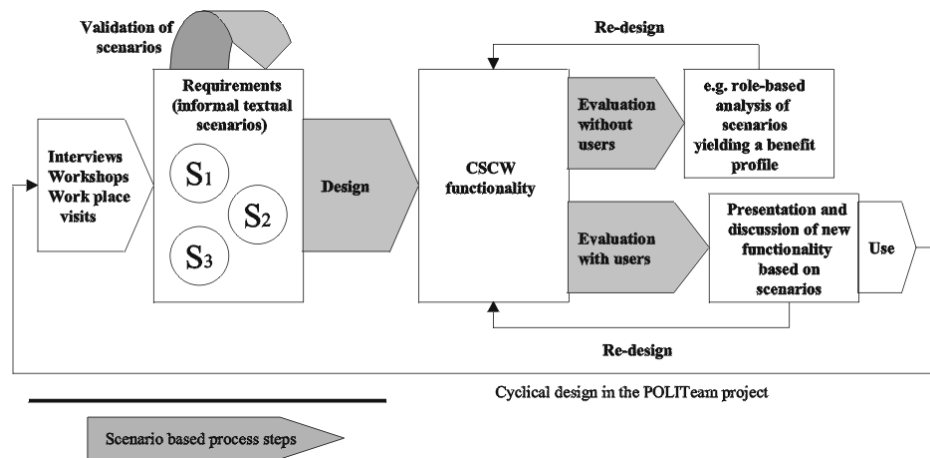


Fig. 1. Scenario support in the POLITeam design process.

munication support) and among designers. This first representation can later be augmented with alternatives, especially visual techniques.

2.2 Building Cooperation Scenarios

The information captured in cooperation scenarios is gathered in the POLITeam project through extensive field studies, involving semistructured interviews and work place visits. While these techniques are usually single user oriented, we attempt to identify the role a person plays in the organizations and the different cooperative tasks. The question of individual motivations and goals is especially important in the context of groupware development, as the product has to be acceptable to users in radically different roles with respect to the cooperative activity which is being supported by the system (e.g., managers and subordinates, see [10]). For this purpose, we use what we call *continuously refined heuristic user selection schemes* (see [25]). Having identified a cooperative activity, we attempt to identify the users involved in this activity. For the interviews and work place visits, we select users who play different roles in the cooperation. However, our initial selection of users is often not exhaustive and as we learn more about the processes in our target organizations, we identify additional users, who might play a different role in a more subtle, however relevant, way. Thus, we continuously refine the user selection.

A methodologically important aspect of cooperation scenarios is the degree to which the cooperation in the respective field of application is already supported by computer systems. The three scenarios described in the next section range from full computer support to completely non-IT based activities which might be subject to future groupware support. The existing work practices are usually heavily influenced by the features (and problems, see scenario one) of the computer system already in place. It is essential to abstract from existing technology in order to understand the nature and especially the motivation and goals behind the cooperative activity. The main point of this article is to make a case for the value of scenarios as support for developing this understanding. Additionally, we will discuss the value of scenarios in the evaluation phase.

3 THREE EXAMPLE SCENARIOS CONCERNING ACCESS CONTROL IN THE CONTEXT OF COLLABORATIVE ACTIVITIES

When the first POLITeam version was introduced, it quickly became obvious that the users had severe problems with the traditional (matrix based) access control system (see e.g., [20]). They just did not use it. In order to understand the nature of these problems, we began to investigate the work practices revolving around access to collaboratively used documents.

The three scenarios presented in this section were distilled from a series of interviews in two fields of application of the POLITeam project and an additional interview session at a newsletter's editorial office, together with a workshop with users from the federal ministry involved in the POLITeam project. The workshop was also concerned with other questions (which are not relevant here) and is not to be confused with the feedback workshop described later on in this paper. The scenarios and their analysis are taken from [26], where they are used to motivate the novel aspects of the access control system. (The main points of [26] are technical aspects of the implementations of this system and thus it employs a more formal method than scenarios (Petri nets) in order to describe and discuss implementation details and alternatives at the heart of the novel functionality. Stiemerling and Wulf [26] do not discuss the design process as such).

QUOTED TEXT

First Scenario. A state representative body—keeping user passwords in sealed envelopes in a strongbox.

This scenario is based on the working practice of the representative body of a North German state at the federal capital in Bonn. As participant in the POLITeam project, the body has been equipped with a groupware system, supporting internal cooperation as well as the cooperation with the state government in the state capital. The installation consists of 28 workstations. An important cooperative task being supported is the preparation of documents for the state's vote in the Bundesrat (the German assembly of state's representatives). The cooperative activity is rather time constrained, as decisions concerning the state's vote have to be coordinated, distributed, and validated within a very short time frame (about one week).

In order to deal with unexpected absences due to illness or travel, the office has implemented a seemingly complex, but rather effective non-IT-based work practice around the POLiTeam groupware system. Every user has to write his password on a piece of paper, which is placed in a sealed envelope. This envelope is locked away in the office strongbox. The keys to the strongbox are kept by two trusted persons: the system administrator and the department head (in the following referred to as key-holders).

If a person urgently needs to access a document on the "virtual desk" of an absentee, the person has to ask one of the key-holders to open the strongbox and release the envelope with the password. The envelope is opened and the "virtual desk" can then be accessed with the appropriate password.

On one hand, this work practice is effective in the sense that misuse or illegitimate access is very difficult because of the necessity to negotiate with the key-holder to release the envelope and because the broken seal of the envelope is an indicator that the virtual desktop was accessed.

On the other hand, the granularity of the access granted by the system is rather coarse since once the password is released, any possible action can be taken in the name of the absent person. Documents other than the one needed can be accessed or deleted; mails can be sent and received. Additionally, the eventual change of a user password causes some organizational overhead, because the password not only has to be changed in the system, but also a new envelope has to be placed in the strongbox.

Theoretically, a person could use several virtual desktops with different passwords to achieve a finer granularity. This solution, however, would severely decrease the effectiveness of the regular use of the system since the users would have to log in and out of the system if they change from one task to another. □

Second Scenario. A federal ministry—searching your colleagues' desktops under the watchful eyes of a trustworthy third person.

A different scenario was discovered in a department of a German federal ministry. The department also participates in POLiTeam. The groupware application is used by 12 employees mainly in an operative section and the central typing office. The application supports the cooperative generation of documents, typically involving the head of the section, a member of the section and several typists. In the course of their work, the members of the section occasionally have to search for documents on other colleagues' virtual desktops, for instance, if they need a specific document but do not know who is working on it right now.

For this task the POLiTeam base system provides a search tool which allows users to search for and access all documents which are not explicitly declared private. This tool, however, was severely constrained in the installed version of the groupware system because, in the initial requirement analysis, the users objected to the tool due to privacy considerations. The current version of the tool can only search the folder-hierarchy on one's own virtual desktop.

Conventions have been established which deal with the problem of searching for work outside one's own virtual desktop. Users are requested to keep documents which are

jointly created in shared workspaces. Each member of the section has a workspace whose access he shares with the head of the section and the typists. Thus, the two typists working for the section have a link to all shared workspaces on their desktop. This means that they can conduct a department-wide search for documents. If one of the workers in the section looks for a specific document outside his own desktop, he asks them to search for him and provide him with a link (a POLiTeam concept for shared access to documents) to the found document.

The advantage of this work practice is that the two typists are aware of every department-wide search and—as they are considered extremely trustworthy—nobody feels that his privacy is being invaded. The disadvantage is that access to documents still has to be anticipated, in order for these documents to be kept in shared workspaces. Furthermore, because the typists actually have to search for and then copy or link the respective documents, this practice results in additional work for them, even though its sole point is that they are aware of the search and can intervene, if they doubt its appropriateness. □

Third Scenario. Newsletter editorial offices—smoothing cooperation with limited trespassing in private domains.

Another example for a successful work-practice from the physical world, which is very difficult to support in classical access control systems, was encountered in the editorial offices of a small newsletter, "European agricultural to interested parties." The seven editors are responsible for up to three European countries each.

Thus, in theory the work of the editors is non-collaborative. However, in practice there is lot of information which concerns transnational aspects, e.g., studies comparing agricultural performance of several EU countries or simply a newspaper article from one country which concerns another country. As a consequence, the editors heavily rely on being kept up to date by their colleagues.

The supporting mechanism for this collaboration is a (physical) circulation system, which is based on a number of open post-boxes in the main hall of the suite of offices. The post-boxes are also used for distributing the normal mail to the editors. If an editor wants to share a document with his colleagues, he simply writes the initials of the respective persons on the document and drops it in one of the post-boxes.

Now, sometimes the editor sharing the information decides afterwards that he needs the document himself again. Or he might tell one of his colleagues about the document over lunch, who might discover that this is exactly the information he has been seeking for weeks and immediately needs to see the document. The document in question, however, still resides in one of the post-boxes in the main hall. Thus, in order to speed up the process, the editor searches the post-boxes of his colleagues. However, he would not dare to remove any other documents or look too closely at private mail or faxes, because the post-boxes are located in the main hall and it is very likely that a colleague might pass by. Again, similar to the last scenario, the editor is respecting a social protocol. According to the interviews this custom is honored even by the last person leaving the office at night. When trying to support this kind of work-

practice in a computer system, one quickly reaches the limit of classical access control system. □

4 ANALYZING AND USING THE SCENARIOS TO GUIDE THE DESIGN PROCESS

We have used the three scenarios described above as the basis for the development of a new access control model. We found cooperation scenarios very useful at this point, because they revealed to us some problems with traditional access control systems in collaborative work practice which we had not seen before.

In the following, we briefly describe three observations (taken from [26]) which indicate why traditional access control mechanisms do not support current collaborative work practice very well. These observations are based on the information captured in the three scenarios and guided the subsequent design process, in particular the integration of negotiation and notification mechanisms into the access control system.

QUOTED TEXT

Observation 1. Trusted third persons play an important role in the three scenarios.

The role of a trusted third person is a central feature in the first two scenarios. In state representative office, the employees put the envelopes with their passwords in the strongbox, well knowing that there are others who have the keys. The key-holders are trusted not to abuse their powers. Similarly in the federal ministry the typists are considered trustworthy by their colleagues. Thus, they are allowed to search for documents in restricted spaces. Trusted third persons obviously play an important role when dealing with unexpected absences. This is exactly the point where classical access control systems fail. These systems only deal with two roles, one for the person who specifies the access rules and one for the person who tries to access a document. Access policies, e.g., in form of access lists or matrices, have to be specified in advance, precisely denying or permitting access to an object for certain persons or groups. The problem in the two scenarios is, that the users would like unexpected accesses not to be decided upon in advance, but by a trusted third person in the context of the actual access situation, e.g., by the typists or the key-holder.

In theory, these scenarios could be supported by a traditional access control systems by just granting full access to the trusted person in advance. This solution, however, has the consequence that the trusted person would not only be burdened with the decision concerning access but also has to do the actual work, i.e., accessing and copying the document for the person requesting the access as in the scenario from the federal ministry. If unanticipated accesses are not exceptional, this solution is not acceptable for the trusted persons. Thus, it makes sense to look for a more efficient implementation of these requirements within the access control system. □

Observation 2. Awareness can be used to control access.

Awareness plays an important role in all three scenarios. In the state representative office, the passwords are sealed in envelopes. A broken seal indicates that the password was used. Thus, there is a degree of awareness about accesses to one's desktop. In the federal ministry, access to restricted spaces has to be made aware to the typists. This awareness ensures that every access is carefully considered because one can be held accountable for every action. The situation in the editorial offices is rather similar. The position of the post-boxes in the main hall provides a degree of awareness, which ensures that other editors' boxes are only accessed if there is a justifiable reason.

Awareness plays an important role by supporting and enforcing social protocols or conventions. Again this is the point where classical access control systems fail. They only allow for the options *yes* and *no*, not *yes, but I want to know about it* or *yes, but I want the typists to know about it*. The POLITeam base system includes functionality which allows users to register interest in changes to specific documents. This functionality, however, is much too coarse in that it does not differentiate between different kinds of accesses. Additionally, it can only be specified for one document at a time, which induces a lot of overhead when trying to use the functionality in real work practice. □

Observation 3. Access can be subject to negotiation at the time of access.

Looking at the work practices described in the scenarios, it is obvious that there is a great need for negotiation. If it is, for any reason, not possible to precisely specify access rights in advance, the anticipated specification is replaced by negotiation during the actual situation of use, either with a trusted third person (see first subsection) or the owner of the document himself. In our scenarios, the necessity for negotiation is a direct consequence of the application of trust in the respective organization.

In the scenario taken from the state representative body, one has to negotiate with one of the key-holders about the release of the password. The key-holder has the opportunity to grant or deny access according to his understanding and motives in the current context. In the federal ministry one has to negotiate with one of the typists in order to search for certain documents. □

4.1 Implementation

Based on this analysis, we have developed a new access control model, which integrates awareness and negotiation services with a traditional, anticipative access control system. When specifying access rights for a document or a shared folder, the users not only have the opportunity to allow or deny access in advance. They have two additional options: *notification* and *negotiation*. The former allows access, but notifies a single user or a group about the access. The latter only allows access, if a specified single user or group agrees. For a detailed discussion of the implementation see [26].

5 SCENARIO SUPPORTED EVALUATION

So far, we have described the use of cooperation scenarios for the purpose of capturing requirements which stem from cooperative work practice and guide the design of new CSCW functionality. However, we also found scenarios very useful during the evaluation phase. As mentioned in the introduction, evaluating groupware is rather difficult, due to the many factors involved in success or failure of a product. A full-fledged test with groups of users—preferably at their work place—is impractical before the product has reached a certain maturity (especially stability). If groups of users are invited to the lab, they leave a lot of important context factors behind and are subjected to several new ones. Our strategy is to capture as much of the relevant context as possible in cooperation scenarios, especially the motivation, the goals, and also the workload of the different participating roles. We try to evaluate our designs against this information as early as possible in the process. Specifically, we use scenarios for evaluation in three stages of the design process (see Fig. 1): evaluation of scenario validity, theoretical evaluation of system design, and the practical evaluation in user workshops.

5.1 Evaluation of Scenario Validity

When we transcribed the scenarios presented in [26], we found that there were some differences in the interpretation of relevant details. This was despite the fact that both authors had attended the (first) workshop (which had been protocoled) and both authors conducted the relevant interviews together (which had been audio-taped).

A good example for a significant misinterpretation was discovered when writing up and discussing scenario two. During the workshop, one of the attendees from the federal ministry was talking about searching other peoples desktops and the importance of other, trusted persons knowing about this search. She was referring to the “virtual” desktops of her colleagues, which was clear to everybody knowing that she had her office three stories above the colleagues she was talking about. However, the author writing up the workshop protocol did not know that and assumed she was talking about “physical” desktops. The mistake was finally discovered, when the other author reviewed the scenario.

We believe that misinterpretations like this are a major problem in design processes with user participation. Common misconceptions can be cemented into what we call “project folklore,” i.e., things that all designers believe in and which even are passed on to new project members as “facts.” Our cooperation scenarios rely on anecdotal evidence which can be severely distorted depending on who writes it up and what prior knowledge this person has about the field of application in question. Thus, we found it very helpful, to exchange cooperation scenarios among designers after write-up and critically compare and discuss them, specifically asking questions like “*I did not get this detail. Who said that, and are you sure you’ve interpreted it correctly?*” As a last measure, we sometimes even asked the interviewees later on, whether this or that detail matched reality. For this purpose a feedback workshop with all end users involved is quite useful.

5.2 Theoretical Evaluation of System Design

As mentioned before, we tried to capture not only the cooperative activity but also the motivation and goals behind individual contributions to the activity. Another important aspect is the workload of a specific role in the scenario.

Having produced an early stage of the new design, we can already employ scenarios for a role-oriented benefit evaluation. We “insert” the new design, i.e., the new CSCW functionality, in the scenario and analyze for each role in the cooperative activity how individual parts of the task change. Take, for instance, the first scenario. In the old scenario, the system administrator or department head had to be asked to open the strongbox and extract the password from the envelope. In the new scenario, the documents are protected employing the negotiation service. How does the new scenario impact upon the individual parts of the tasks? Table 1 shows a role-oriented analysis of the old and new scenario one.

Role-oriented analysis of technology-enhanced scenarios can show, who benefits from the introduction of new technology and who has to carry additional burden (compare [10]). The analysis shown in Table 1 indicates that the user requesting access and the system administrator benefit from the new functionality, while the data owner has additional work in specifying the configuration of access rights and negotiation services.

While role-oriented analysis gives some hints concerning the possible acceptance of a system, there are still a lot of factors which might be relevant, but do not surface in the analysis. For example, it might be a huge benefit for the data owner for personal reasons, if he can specify another person than the system administrator as decision maker. Or there might be certain documents on a persons desktop which have to be excluded by law from the scheme (e.g., medical documents), which is possible with the fine-grained configuration of the negotiation service. These examples show the importance of capturing context in cooperation scenarios for theoretical evaluation.

5.3 Practical Evaluation of System Design in Workshops

While theoretical analysis might yield some useful hints concerning possible problems with the groupware system, the almost canonical unreliability of any requirements analysis process indicates the need for user feedback as early as possible in the design process.

For this purpose we conducted a feedback workshop, during which we presented an early (rather unstable) version of our prototype to end user. (The feedback workshop is not to be confused with the earlier one.) Eleven end users from one of our fields of application (the federal ministry) participated in the workshop, among them the head of department. Seven designers were present; three from the University of Bonn and four from GMD, St. Augustin, Germany, where the workshop took place. The project members from GMD were not directly involved in the design of the access control system presented here, but contributed to the discussion during the workshop. The main goal of the presentation of the prototype was to help the users in understanding the new functionality achieved by

TABLE 1
ROLE-ORIENTED ANALYSIS OF THE NEW TECHNOLOGY-ENHANCED SCENARIO ONE

Role	Old scenario	New scenario with negotiation service support
User requesting access	<ul style="list-style-type: none"> • Has to contact system administrator. • Has to wait for system administrator's decision • Has to log in under data owner's account • Has to copy the relevant documents to his own account • Can be held responsible for any additional changes to data owner's account 	<ul style="list-style-type: none"> • Has to wait for system administrators decision (System administrator or any other trusted person is contacted automatically)
System administrator (department head is omitted here for simplicity)	<ul style="list-style-type: none"> • Has to make decision • Has to open safe • Has to extract password from envelope • Has to communicate password to user • Has to change password later on • Has to create new envelope with new password 	<ul style="list-style-type: none"> • Has to answer the automatically generated request.
Data owner	<ul style="list-style-type: none"> • Does not know, what other data has been compromised • Has to submit his password to the system administrator in advance 	<ul style="list-style-type: none"> • Has to specify access rights and negotiation service in advance (can be a lot of work in dynamic work environments)

our integration of traditional “yes or no” access control with negotiation and notification services. Based on this understanding we wanted to know from the users how they believed the system could be used to support (or change) their work practices. Implementation details (e.g., the user interface) of the prototype were also discussed, but are not in the scope of this paper, as are the usability tests we ran (not during the workshop) to evaluate the interfaces for configuring the negotiation and notification services.

The presentation during the workshop was based on a scenario drawn from the federal ministry concerning the deletion of an address list in a common workspace (not one of the scenarios described before). Together with student volunteers we “played” the technology enhanced scenario from a script, which we had ensured did not cause the prototype any trouble. The screens of two workstations were fed into beamers, so all participants could see the details. The workshop, especially the discussion and contribution of the end users, was protocoled.

The real world cooperation scenario captured the imagination of the workshop attendees and helped us, first to discover some flaws in the base scenario and secondly to discuss the design of the new functionality (the users wanted more high-level and powerful configuration mechanisms for the negotiation service and the basic access rights, which supports the result from the theoretical analysis that the additional workload for the data owner might be a problem). Concerning the central question of how the new functionality might support their work practices, the users voiced rather diverse opinions. Interestingly, the head of the department said that if all responsibilities (e.g., for deleting address lists) were assigned “correctly,” negotiation during the time of access would not be necessary. Furthermore, a high level of individualized (as opposed to standardized) procedures would actually hinder coopera-

tion, because single contributions would not fit together. Both points were rather vehemently opposed by his subordinates, who stated that there were not one “correct” way to assign the responsibilities, but that in their everyday work practice many things were in flux and could not be cast into fixed and standardized structures. From each point of view, the statements are justified. The head of department wants to have clear (at best static) picture of what is going on in his department, while the subordinates have to actually deal with the surprises and nonstandard situations in everyday work. This exchange of views exemplifies the group-related aspects which were mentioned in the introduction and which CSCW design has to take into account. The cooperation scenario we employed as basis for the presentation, helped in supporting the understanding of the end users and the subsequent discussion.

6 RELATED WORK

The approach presented here mainly builds on work which is concerned with putting the end user and his or her work in the focus of attention during software development. This perspective is due to the nature and the current state of CSCW. The discussion in this field is still in an exploratory phase. It is concerned with the question of what functionality groupware systems should provide (e.g., notification mechanisms) and what the benefits of these systems for individual users, groups of users and organizations really are. This reflects on the design methodologies which are used to support groupware design processes. While more traditional software design (see e.g., [24]) proceeds rather quickly from the initial requirements analysis and specification phase to formally verifiable application models and perhaps executable specifications, the designer of a groupware system has to spend much more time in the early

phases of development, envisioning how the new functionality might change current work practice in an organization. Similarly, Hammer and Champy ([11]) postulate the “*enabling role of information technology*” (p. 83). Organizational and technical changes are intertwined and cannot be viewed or implemented separately. The introduction of a groupware system should not cement existing work practice in program code, but allow for new forms of collaboration which perhaps better support the organizational goals.

Putting the end user and his or her work in the focus of attention is not a new concern in system development. As computers enter more domains of private and working life, novel uses of information technology are pioneered. This development necessitates design methodologies which put the use and the user of the system in the center of the development efforts. This does not mean that traditional, more formal techniques are obsolete. To the contrary, as systems become more complex, formal methods are indispensable. However, they have to be complemented with a sound understanding of the current work practice and the possible future use of the system from the end users’ perspective.

The *use-case* methodology by Jacobsen et al. ([15]) is an example for the incorporation of the end user’s perspective into system design. *Use cases* are descriptions of the interaction of an actor external to the system (usually a human user) and the system itself. A use case thus encapsulates “*one specific way of using the system by using some part of the functionality*” (p. 154). A set of uses cases and the set of related actors (or user roles) constitute the *use case model* which is part of the whole requirements model (together with a problem domain object model and user interface descriptions). Use cases thus permit the explicit representation of the intended use of the system and can serve as a basis for discussion with the end users. In this respect they are similar to our technology-enhanced scenarios. However, as use cases represent interactions of single users with the system, the cooperative aspects of using the system are not adequately captured. While it is possible to decompose the cooperative use of a groupware system into several use cases related to different actors, this decomposition omits—especially in the case of synchronous groupware—essential design information like dependencies between different use cases which actually concern the same cooperative system usage. We have made the experience that groupware designers are prone to neglect the cooperative aspects of system use when specifying requirements as use cases.

In order to capture the complex dependencies and requirements arising from cooperative work practices, CSCW often draws on methods from the field of Participatory Design (PD). These methods are based on the actual end users taking part in the design process of the system. The first PD projects in Scandinavia were trade union-oriented and focused on giving workers the right to influence their own working conditions, as well as on improving the design of computer systems (For an overview of the history of PD see e.g., the introduction to [8]). However, the obvious advantages of letting those people who know most about the work participate in the design of the systems supporting it, give the methods developed in the context of PD a pragmatic significance beyond their initial political coloring

([16]). PD approaches usually focus on the first part of development processes and spend a lot of time building an understanding of the work context and current work practices of the end users. Sometimes they initially employ ethnographical techniques as a way of gathering socially oriented context information (see e.g., [2]). These techniques rely on project members working very closely with the users in their “natural” surroundings. Not unlike the observation of a tribe’s social structures and rules in the Amazon, project members study fields of application like air traffic control centers ([2]) or a control room for the London Underground ([12]). However, ethnographical studies in the proper sense take a lot of time (several month, perhaps years). Thus, their usefulness for commercial (or even short term research) projects has been doubted. Hughes et al. [14] discusses different roles which ethnography can play realistically in a design process, ranging from short term evaluation of prior, more long term studies to “quick and dirty” ethnography with the goal of gaining as much insight as possible in a short time. In Holtzblatt and Beyer’s *Contextual Design* methodology [13], the “*contextual inquiry*” phase ([13, p. 93-94]) is an example for the pragmatic use of ethnography in design projects. They suggest that project members—at best the actual designers—spend time (2-3 hours per session) with the users in their usual environment while they are working. The project members can interrupt the work at any time and ask about the goals and motivations behind user actions. In order to provide an efficient view across the whole organization, several project members interview several users in parallel. The interview results are later shared within the project team during a process of structuring what is known about the design problem. Additionally the method employs a variety of work models to capture e.g., the single tasks, steps, or the strategy of the work to be supported. Relevant for CSCW, for instance, is the *flow model* (p. 97) which depicts the communication between people in the organization. However, the method does not impose a single modeling language on the designer, but proposes to introduce new languages if needed, since there is no one language which can capture all relevant aspects: “*Let modeling languages help you. When you must, invent new ones to say exactly what you have to say*” (p. 96). Contextual design also involves iteratively going back to the users with clarification questions and design suggestions, e.g., in form of rough paper prototypes (see e.g., [5]) of user interface designs. Even though contextual design does not rely on scenarios in the way the approach presented in this paper does, it is based on a similar understanding of design of interactive systems as a creative and user-centered process.

Another approach, which is very much in the tradition of PD, is the *MUST* method by Kensing et al. ([16]). *MUST* is a Danish acronym for theories of and methods for initial analysis and design activities. *MUST* specifically supports design in an organizational context and thus explicitly includes steps like *strategic analysis* which integrates the development efforts into the overall business strategy of the organization. The approach suggests a variety of techniques for developing an understanding of current work practices, including interviews, observations, workshops, document

analysis of documents used in the work practice etc. The method also places a strong emphasis on the cooperative development of visions of the future system by both users and developers. Within this process, the use of “*scenarios describing envisioned future work practice supported by the proposed design*” (p. 137) is suggested. Concerning the use of formal methods, the authors state: “... *formalism play a minor role in the MUST method. Instead we suggest plain text, freehand drawings, and sketches for the production and presentation of the relation between proposed IT systems an users’ current and future work practice, postponing an extended use of formalism to later on in the development process*” (p. 138). Because of its strong organizational focus and its reliance on the PD tradition, the MUST method attempts to explicitly accommodate conflicts of interest between management and workers e.g., in rationalization and downsizing processes. The authors suggest to achieve a consensus concerning the objectives of the system design beforehand. While these issues are out of the scope of this paper, it is nevertheless interesting to note how far the consequences of system design can reach and what responsibilities result for the designers. Scenarios can serve as a tool to make these consequences explicit for everybody involved in the process.

A concrete application of scenarios in a CSCW project is discussed by Kyng ([19]), who describes the use of a variety of scenario types in the context of the EUROCOOP project. This project was concerned with designing computer support for cooperation in the Great Belt Link Ltd. Company, a state-owned company responsible for the building of a bridge/tunnel between Zealand and Funen in Denmark. It involved the design of four generic, interrelated CSCW applications. Scenarios were used in four different roles: *work situation descriptions* were supposed to capture relevant, existing situations which the users find to be important parts of their work, bottlenecks, or generally insufficiently supported. These freeform textual descriptions (which Kyng does not yet call scenarios, in contrast to the terminology of this paper) first served as basis for the discussion of current work practices between users and designers. Secondly, they were used in the process of developing mock-up prototypes and accompanying *use scenarios* which textually describe the intended future use of the envisioned system. In this sense, use scenarios are similar to our technology-enhanced scenarios. In the EUROCOOP project, however, they were mainly employed in setting the stage for user workshops and not for theoretical evaluation. Kyng also describes the project-internal use of more detailed, technically oriented *exploration/requirement scenarios*. They complemented use scenarios by giving details which are relevant for evaluating technical details (e.g., locking mechanisms) of the proposed design. Finally, *explanation scenarios* gave a description of the new possibilities offered by the proposed design and the explanation of the rationale behind the design in terms of the working situation. These scenarios were more detailed than use scenarios.

Building on the work presented by Kyng, Bardram ([1]) describes the use of scenarios in the SAIK project which was concerned with the design of a Hospital Information System in Denmark. Similar to our work, scenarios were used to describe the current work practices and to envision fu-

ture, computer supported work activities. Bardram’s approach is characterized by the fact that scenarios were used on two different levels of detail. All relevant work practices were textually described as *work activity scenarios*, while work activities central to system design were documented in greater detail with the help of more structured, tabular *analytical scenarios*. Furthermore, Bardram points out that current and future scenarios were “*alive during the whole system development process*” (p. 59) and were constantly updated as the designer’s understanding of current work practices and the design itself evolved. In contrast to the role of the use scenarios described by Kyng, analytical scenarios were also employed in a systematic comparison (*logical confrontation* in Bardram’s terminology) of the current work practice and the proposed design. This comparison resulted, for instance, in the discovery of problems concerning the integration of the new system with existing systems for the exchange of EDIFACT messages with external agencies. Additionally, Bardram advocates the use of scenarios in workshops together with the future users and management representatives (compare [16]).

This overview of related work shows the span of different roles scenarios can play in the design of systems supporting cooperative work. Scenarios are used to describe current and future, technology supported work practices. They are used for internal, technically-oriented discussions among designers and as a basis for the presentation of prototypes of future systems to end users. They can have different levels of detail and different forms (e.g., tabular or textual). Furthermore, scenarios can be used to support design processes which accommodate organizational and business strategy issues as well as processes which focus on the differences the system makes for individual users (compare our role-based analysis). We conclude that scenarios are an extremely flexible tool for system design and that there is not one single best form and use of scenarios. Depending on the circumstances and constraints of the project, the preferences and experiences of the designers, and last but not least the end users, scenarios can be employed in a number of variations.

7 SUMMARY AND DISCUSSION

We have presented our use of cooperation scenarios in the design and evaluation of a novel access control system for groupware. Cooperation scenarios are context-rich, informal, textual descriptions of cooperative activities which are gathered and refined through interviews and workshops with user participation. They not only contain a step-by-step description of events but also the goals and subjective opinions (e.g., trust) of persons and other, possibly relevant contextual elements. During evaluation we have used scenarios for requirements validation (through communication among designers), role-oriented analysis, and as a basis for realistic workshop presentations.

The main advantage of cooperation scenarios lies in the fact that at least some contextual factors can be taken into account early on in the design process. Other, more restrictive forms of scenarios or formal models might omit this qualitative and sometimes rather subjective information.

We have shown how cooperation scenarios can be used to identify issues which are important for cooperative activity (trust, awareness, and negotiation). The scenarios allow the designers to think in terms of actual use situations instead of abstract technical criteria. Especially the role-oriented analysis allows a multi-perspective view on a possible design, perhaps helping to identify problems early on (e.g., a benefit and workload disparity). Furthermore, we found that realistic workshop presentations make it possible to establish close contact with users and get them involved in the evaluation and generation of new ideas in the early stages of the project.

On the negative side, we were surprised how easily anecdotal evidence becomes severely distorted due to misunderstandings and misinterpretations of issues in the field of application (e.g., the "virtual"/"physical" desktop problem) and how far apart the different designers' ideas about the users and the field of application can be. We have suggested addressing this problem by critically reviewing scenarios among designers and validating them in feedback workshops.

Furthermore, we point out again that we see the role of cooperation scenarios early on in the design process when creating innovative functionality and envisioning its effects on cooperative work. Our review of the current literature shows that this reflects the experience made with the use of scenarios in other (CSCW) projects. A good system design subsequently of course makes more formal methods necessary to produce a high quality implementation.

8 FUTURE RESEARCH

We still have a lot to learn about how to design CSCW functionality and how to introduce groupware systems into organizations. Experiences from our projects and reports in the current literature (see e.g., [21]) show that *tailorability* (or adaptability) is an important success factor for such system because of the diversified and dynamic nature of requirements of cooperative work. Most current design methodologies still aim at producing a "one-size-fits-all" system design. Part of our current work is concerned with the development of methodologies which not only explicitly specify diversity and dynamics but also help us in deriving the necessary degree of flexibility the final system has to exhibit.

ACKNOWLEDGMENTS

We thank the other past and present members of the POLIteam project at the University of Bonn: Andreas Pfeifer, Helge Kahler, Volkmar Pipek, Markus Won, and Volker Wulf. Furthermore, we are grateful to our POLIteam project partners at GMD in St. Augustin, Germany. Furthermore, the comments of the anonymous reviewers were extremely helpful in improving the final version of this article. The POLIteam project is funded by the BMBF (German Ministry of Research and Education) in the context of the PoliKom research program under Grant No. 01 QA 405/0.

REFERENCES

- [1] J.E. Bardram, "Scenario-Based Design of Cooperative Systems," *Proc. COOP '98*, vol. 1, pp. 57-68, F. Darses and P. Zaraté, eds. Cannes, France: INRIA, 1998.
- [2] R. Bentley, T. Rodden, P. Sawyer, and I. Sommerville, "Ethnographically-Informed Systems Design for Air Traffic Control," *Proc. CSCW '92*, pp. 123-129, J. Turner and R. Kraut, eds., Toronto, Canada: ACM Press, 1992.
- [3] J.M. Carroll, *Scenario-Based Design*. New York: John Wiley & Sons, 1995.
- [4] P. Dourish and V. Bellotti, "Awareness and Coordination in Shared Workspaces," *Proc. CSCW '92*, pp. 107-114, Toronto Canada: ACM Press, 1992.
- [5] P. Ehn and M. Kyng, "Cardboard Computers: Mocking-It-Up or Hands-On the Future," *Design at Work*, J. Greenbaum and M. Kyng, eds. Hillsdale, N.J.: Lawrence Erlbaum Assoc., pp. 169-196, 1991.
- [6] C.A. Ellis, S.J. Gibbs, and G.L. Rein, "Groupware—Some Issues and Experiences," *Comm. ACM*, vol. 34, pp. 38-58, 1991.
- [7] L. Fuchs, U. Pankoke-Babatz, and W. Prinz, "Supporting Cooperative Awareness with Local Event Mechanisms: The GoupDesk System," *Proc. Fourth European Conf. Computer-Supported Cooperative Work*, pp. 247-262, H. Marmolin, Y. Sundblad, and K. Schmidt, eds., Stockholm, Sweden: Kluwer Academic, 1995.
- [8] J. Greenbaum and M. Kyng, *Design at Work*. Hillsdale, N.J.: Lawrence Erlbaum Assoc., 1991.
- [9] I. Greif and S. Sarin, "Data Sharing in Group Work," *Proc. CSCW '86*. Austin, Texas, 1986.
- [10] J. Grudin, "Groupware and Social Dynamics: Eight Challenges for Developers," *Comm. ACM*, vol. 37, pp. 93-105, 1994.
- [11] M. Hammer and J. Champy, *Reengineering the Corporation—A Manifesto for Business Revolution*, first edition. New York: Harper-Collins, 1993.
- [12] C. Heath and P. Luff, "Collaboration and Control: Crisis Management and Multimedia Technology in London Underground Line Control Rooms," *Computer Supported Cooperative Work (CSCW), An Int'l J.*, vol. 1, pp. 69-94, 1992.
- [13] K. Holtzblatt and H. Beyer, "Making Customer-Centered Design Work for Teams," *Comm. ACM*, vol. 36, pp. 93-103, 1993.
- [14] J. Hughes, V. King, T. Rodden, and H. Anderson, "Moving Out from the Control Room: Ethnography in System Design," *Proc. CSCW '94*, pp. 429-440, R. Furuta and C. Neuwirth, eds. Chapel Hill, N.C.: ACM Press, 1994.
- [15] I. Jacobsen, M. Christerson, P. Jonsson, and G. Övergaard, *Object-Oriented Software Engineering—A Use Case Driven Approach*. ACM Press, 1992.
- [16] F. Kensing, J. Simonson, and K. Bødker, "MUST—A Method for Participatory Design," *Proc. PDC '96*, pp. 129-140, J. Blomberg, F. Kensing, and E. Dykstra-Erickson, eds. Cambridge, Mass.: ACM Press, 1996.
- [17] K. Klöckner, P. Mambrey, M. Solenkamp, W. Prinz, L. Fuchs, S. Kolvenbach, U. Pankoke-Babatz, and A. Syri, "POLIteam—Bridging the Gap between Bonn and Berlin for and with the Users," *Proc. ECSCW '95*, pp. 17-32, H. Marmolin, Y. Sundblad, and K. Schmidt, eds. Stockholm, Sweden: Kluwer, 1995.
- [18] K. Kuutti, "Work Processes: Scenarios as a Preliminary Vocabulary," *Scenario-Based Design*, J.M. Carroll, ed., pp. 19-36. New York: John Wiley & Sons, 1995.
- [19] M. Kyng, "Creating Contexts for Design," *Scenario-Based Design*, J.M. Carroll, ed. pp. 85-108. New York: John Wiley & Sons, 1995.
- [20] B.W. Lampson, "Protection," *ACM Operation System Rev.*, vol. 8, pp. 18-24, 1974.
- [21] T.W. Malone, K.-Y. Lai, and C. Fry, "Experiments with Oval: A Radically Tailorable Tool for Cooperative Work," *ACM Trans. Information Systems*, vol. 13, pp. 177-205, 1995.
- [22] H. Oberquelle, "Situationsbedingte und benutzerorientierte Anpaßbarkeit von Groupware," *Menschengerechte Groupware—Software-ergonomische Gestaltung und partizipative Umsetzung*, A. Hartmann, T. Herrmann, M. Rhode, and V. Wulf, eds. pp. 31-50, Stuttgart, Germany: Teubner, 1994.
- [23] H. Shen and P. Dewan, "Access Control for Collaborative Environments," *Proc. ACM Conf. Computer Supported Cooperative Work (CSCW '92)*, pp. 51-58, Toronto, Canada: ACM Press, 1992.
- [24] I. Sommerville, *Software Eng.*, third edition. Reading, Mass.: Addison-Wesley, 1989.
- [25] O. Stiemerling, H. Kahler, and V. Wulf, "How to Make Software Softer—Designing Tailorable Applications," *Proc. DIS '97*, G.v.d.

Veer, A. Henderson, and S. Coles, eds., pp. 365-376, Amsterdam: ACM Press, 1997.

- [26] O. Stiemerling and V. Wulf, "Beyond Yes or No'—Extending Access Control in Groupware with Negotiation and Awareness," *Proc. COOP '98*, vol. 1, pp. 111-120, F. Darses and P. Zaraté, eds., Cannes, France: INRIA, 1998.
- [27] V. Wulf, "On Conflicts and Negotiation in Multiuser Application," *Encyclopedia of Microcomputers*, A. Kent and J.G. Williams, eds., New York: Dekker, 1997.



Oliver Stiemerling received the degree of Diplom-Informatiker from the University of Bonn in 1996, after studying computer science and business in Bonn and at the University of Warwick, Great Britain. He worked in several groupware projects in industry and university, concerning the computer support of cooperation in public administration, engineering companies, and IT support departments. He is currently working as a researcher in the Department of Computer Science III at the University of Bonn.

His research interests are CSCW, tailorability, design for change, component software, formal component models, and access control. He is a member of the Association of Computing Machinery.



Armin B. Cremers received the degree of Diplom-Mathematiker, the Doctor of Science degree in mathematics, and the *venia legendi* in informatics from the University of Karlsruhe in 1971, 1972, and 1974, respectively. In 1976, after a stay as assistant professor at the University of Southern California, Los Angeles, he was appointed professor of informatics at the University of Dortmund, where he taught data structures and database system design. Since 1990 he represents the fields of information

systems, software technology, and artificial intelligence at the University of Bonn. He is a member of the Association of Computing Machinery.