In *Proceedings of the ACM 1998 Conference on Computer Supported Cooperative Work (CSCW'98)*. New York: ACM Press, pp. 373-382.

Designing Groupware for Congruency in Use

Wolfgang Prinz, Gloria Mark, Uta Pankoke-Babatz
GMD - FIT
Schloß Birlinghoven
53754 Sankt Augustin, Germany
{wolfgang.prinz, gloria.mark, pankoke}@gmd.de

ABSTRACT

In this paper, we present experiences from long-term groupware development, introduction, and use in an organization. We report lessons learned concerning how a complex design process operates and how its components interact. Our experiences suggest that the processes of requirement analysis, system development, and user support need to facilitate the merging of individual work patterns into congruent system usage. We confirm the changing nature of groupware use by reporting empirical results describing different learning phases.

Keywords

Cooperative design, user advocacy, groupware requirements, group work, system introduction, prototyping.

INTRODUCTION

Observing the process of groupware use as it unfolds over time can reveal a number of subtle and important factors that influence how progress is made. A longitudinal observation also enables the reconstruction and evaluation of decisions concerning the design process and system development.

Despite the widespread use of groupware systems, the process of groupware design and use is still not well understood. From the perspective of system development, observation over time has provided some valuable insights, e.g. in identifying a time frame for system changes [26]. From the perspective of system use, several important studies have identified factors that groupware practitioners should be aware of when implementing systems: e.g. the extra burden imposed on some group members through groupware participation [21], the imbalance in costs and benefits [7], unanticipated overhead [3], as well as the potential for innovating work practices [15]. Only by observing within a long timeframe could such lessons be learned.

Designing groupware requires an acknowledgement of the interplay between the tasks of gaining design requirements, system use, and system development. These processes interact and evolve over time, each aspect exerting influences on the other components. One of the main lessons learned from our experiences is that these processes need to be directed toward the goal of supporting system usage by group members so that their actions are congruent with each other. Congruency refers to all members having the same assumptions of how the system is to be used in performing their work. This cannot be accomplished unless the design team and users achieve a common understanding of the users' tasks, work processes, and system design. Thus from our view, most importantly what we gain by observation over time is the capability to better understand the dynamics of these processes in order to identify lessons and establish guidelines for future practice.

In this paper we contribute towards understanding groupware use and design by presenting empirical results from different phases involved in groupware adaptation and development. The results are based on work in the POLITeam[1] project [8] in which a groupware system was used in the course of daily work in a government ministry. From the beginning of the project to the present, we have had the opportunity to observe the users (and ourselves) through three years of system use, from introduction, through initial growing pains, through coordination difficulties, to the point where the users can report that in retrospect, the work to learn the system has paid off: groupware use has increased the efficiency of their work. As the users developed and improved their understanding of system use, and of their group, so did the design team as well improve its understanding of how the design process and the system itself affects the adaptation and learning process. To this end, in this paper we report lessons learned from the various perspectives involved in groupware implementation, and offer our experiences on what fosters and hinders system use.

This paper begins with a description showing how coordination with groupware use is related to the process of group development. We then outline our particular design approach explaining how we tried to achieve a common understanding between the design team and users, and how we tried to account for user group development.

[1] POLITeam is funded by the German research ministry in the framework of the POLIKOM initiative.

In the following section we present empirical results on the actual use of the system in the work setting, illustrating how the group strived to achieve congruent conventions. Finally, we outline the lessons we have learned during four years of groupware system development in close cooperation with users, including three years of practical use.

ACHIEVING CONGRUENT USE OF GROUPWARE

When we first consider the process of learning groupware, we must take into account not only understanding the system functionality, but as many researchers in CSCW have pointed out, there must also be reflection on how members need to develop procedures to coordinate their work and the use of the functionality [10, 12, 20]. Developing coordination practices goes hand-in-hand with enabling various social processes in the group to develop, e.g. the building of communication networks, role adoption, resolving status and hierarchies, and articulation, to name only a few.

During the course of a group's development, group members slowly merge their different attitudes, behaviors, and social norms, gradually moving towards a congruency in the group [14]. Karl Weick [27, p. 46] describes this development as organizing a collective structure which exists "when behaviors of two or more persons become interstructured". The process of electronic group formation operates in a similar way, with some important differences. The electronic interaction of group members often occurs while members are remote. We refer to achieving congruency with groupware use as involving two processes: first, the group must develop appropriate communicative behaviors, and secondly, group members must merge different perspectives to reach common agreements about cooperation.

Communicative behaviors are the means through which group members can influence each other and create standards of behavior: e.g. by observing each other's actions, or by explicit and implicit communication [28]. The difficulty that electronic groups face during system use is that they lack the social information that groups generally gain through formal and informal interaction. This social information helps group members to evaluate each others' actions and observe discrepancies. Through this, members gradually accommodate themselves to the group (as the group accommodates to individual styles), developing commitments along the way [13]. It is in this accommodation process in which conventions are formed by the group for using the system in cooperative work. The conventions that must be formed are not static; they evolve with the group, must be robust, yet at the same time must be flexible to adapt to local contingencies [12].

In general, in groupware design, the group process and the technical system cannot be considered separately [5, 16]. Because of this interdependence, the design of the system, and the process used to support users and gain design requirements, should function together to facilitate the development of congruent behaviors in an electronic group. Two ways of achieving this are by supplementing user-user communication, and by supporting the process of establishing and maintaining conventions. Above all, the congruency process cannot be supported unless the design team and the users have a common understanding of goals and of the work to be supported.

While it is now more clear for us how the design process and groupware learning can influence each other, in the next sections we report on our design approach which aimed to achieve a common understanding between the design team and users so as to develop adequate user system support and facilitate congruent use of the system.

THE DESIGN APPROACH

The aim of our approach was that concepts and systems for the support of group work need to be developed in a situated manner [23], such that the technical system, the work practice and the users' working habits can progress together. None of these elements could be improved separately. Our technical design was guided by the idea of providing a medium that could adapt to the local contingencies of situated work, as opposed to implementing mechanisms that prescribe actions for work.

The Setting

The idea for POLITeam emerged as a result of the decision of the German parliament in 1992 to move the capital from Bonn to Berlin and to split a large number of government ministries between these locations. This decision increases the need for technological support for intra-ministerial cooperation over a distance of 600 km. The German research ministry launched the POLIKOM program in 1994 to investigate technologies to support cooperation, and to develop appropriate design and introduction methods. The focus of the POLITeam project is on the support of ministerial procedures and coordinated document processing.

Within a German ministry, there exists a ministerial organizational handbook, the GGO (a common set of organizational procedures) which prescribes the cooperation processes via hierarchical lines (department, subdepartments, units) as strongly regulated paper-based processes. However, the representation of these rigidly defined processes into electronic cooperation support systems would not be adequate, due to the situated nature of real work [19, 23].

Most cooperative work in the ministry is performed at the Unit level. Therefore, three representational Units of two departments were selected in which to implement the system: two located in Bonn, and one in Berlin. One Unit in Bonn is a central writing office, the other two Units support political decision and management processes. The tasks for these users include speech writing, program development**,** administration of public institutions and accounts, and correspondence with citizens. Accordingly, the majority of working time is devoted to paper-work. This includes the individual or cooperative production of documents or responding to incoming documents.

To support these tasks, the POLITeam system provides an electronic desktop that integrates standard office applications, email, and shared workspaces [17] to share documents for the support of unstructured document production processes, and electronic circulation folders [18] for the support of structured, sequential cooperation processes along hierarchical lines. The technological infrastructure used for POLITeam is LinkWorks, a groupware platform by Digital.

POLITeam is used by the members and heads of the three Units on a daily basis for individual and cooperative document production. Occasional users of POLITeam are managers and their offices at the department and subdepartment level above the unit level.

Shared workspaces were used cooperatively by two different groups in the ministry: typists in a writing office and members of the ministry Units. The members use the workspace to exchange documents with each other. The electronic circulation folders were used occasionally for the transport of bundles of documents between the Units in Bonn and Berlin, for initiating procedures along the hierarchy and for the individual structuring and organization of the electronic desktop.

Designing the user-designer cooperation process

The Scandinavian school developed methods for design at work [9]; this approach enables users to evaluate system prototypes in work-like situations. We based the POLITeam design process on this idea, but went further to enable users to evaluate a system in real work practice. We could then base further system development on these user experiences. We realized that our goal was ambitious and required responsible and careful modes of cooperation with the users since their primary goal, even when involved in a system development project, is to get their work done. Thus, we had to develop methods to involve the users in the design process, yet at the same time ensure that they will benefit in their work from their participation.

Three means helped achieve this goal: the introduction of a groupware system as a means for the articulation of requirements, the establishment of a permanent communication channel through the role of user advocates, and regular workshops for the exchange of feedback between the users and designers. The groupware system, provided to users almost from the very beginning, gave the users a tangible artifact enabling them to experience its usability and to articulate further needs.

To support the communication and cooperation between the design team and users, we developed a new role, the user advocate, which became the interface between the users and the designers. The user advocate had the following tasks:

- to support the users in working with the system, during regular site visits
- to help the users to coordinate their work and to adapt work-habits
- to observe where the system fits and where improvements were required
- to feed the user requirements back into the design process

To support user-user communication, and user-design team communication, workshops were held at regular intervals. The roles of the workshops were:

- to provide feedback on usage and to articulate further needs within the user group and between users and designers
- to give designers feedback to improve their understanding of the users' work
- to demonstrate and discuss new system functionalities
- to discuss and find conventions for the cooperative use of the system

We have augmented these communication channels with regular interviews which allowed the users to rethink and describe their actual work.

Medium vs. mechanism as technical design guideline

Although the GGO procedures appeared at first as a constraint, users revealed in interviews that they did not consider deviations from the standard GGO procedures as exceptions. This led to providing cooperation support based on the provision of flexible cooperation media instead of prescribed cooperation mechanisms. The result of that approach is a set of combinable cooperation tools. Users can coordinate their work using these tools; the tools themselves do not incorporate a priori any representation of the cooperation process [1, 19].

With a media approach, an analysis of cooperation processes in work practice must focus not only on the overall cooperation, but it must also consider individuals' actions with the shared artifacts. The electronic circulation folder has been realized as a medium to express and modify plans for work sequences in the course of their execution. The shared workspace, as a medium, stores and provides the material for cooperation. Appropriate metaphors were used to express the function of the media to the users and to guide designers during the design process [11]. The presentation and functions of the electronic media were related to their counterparts in the real world. In addition, these media had to be combinable with each other, e.g. it should be possible to store a circulation folder in a shared workspace or to forward an alias to a shared workspace by an electronic circulation folder. Similarly, switching actions between different media had to be easy. The exchange of objects, e.g. documents, between different media had to be easy and should not result in a loss of process-relevant meta-data. Additional aspects that are more related to the configurability of a cooperation media can be found in [1].

Cycles of system development

Improvement of the system occurred in cyclic processes: longer outer cycles where communication occurred between the design team and users, and shorter inner cycles contained within, where communication occurred among the design team members.

Outer Cycle

There were four outer cycles, each of which took about one year. The components of the cycles consisted of installing the system (or a new version), and the presence of user advocates at first on a daily basis, and then weekly, with the purpose to train, observe, support, and gather requirements. The cycle continued with workshops, interviews, the development of a new system version, and then installation of that version at the users' site.

During site visits, the user advocates actively solicited problems and requirements. They examined alternative ways to perform tasks with the system. They also detected different and conflicting user habits and supported the coordination of individual work styles. These discrepant workstyles were discussed at regular user workshops which served as the means to find conventions for system use. They also served to raise new user needs and to discuss new solutions between designers, users, and user advocates, which was important feedback used in the design process of the inner cycle.

Inner Cycle

The inner cycle took place within the design team. In this process, the user advocates argued from the perspective of the users, explaining their requirements, and whether they would possibly benefit from a planned feature. The designers proposed potential solutions. The design was done through prototyping, enabling us to recognize the interdependencies between the different system components. In addition, this allowed the user advocates to make a first evaluation of the features and experiment with them, using typical examples from the users' work. This very often led to another redesign cycle. Thus the design procedure was highly incremental. The inner cycle was completed with laboratory workshops where some users could try the new version, sometimes leading to another redesign. This cyclic procedure reduced the risk of disrupting work practice with an unsuitable system version. We never had to withdraw a version once introduced, although sometimes features needed to be redesigned after some use.

Stages of system development

The continuous system development yielded three consecutive POLITeam versions. The characteristic of each version mirrors the evolutionary stage of user requirements in the course of system use. We introduced the first version of POLITeam in February 1995, the second version in February 1996, the third one in December 1996, and the final one was ready in December 1997.

The first POLITeam version provided a technically stable and functionally sufficient system. In a simple way it provided most of the functionality needed by the users to start with the requirement analysis in the work setting. The functionality of that version was derived from initial user interviews. This version included a personal electronic desktop, email, an electronic circulation folder, and a shared workspace. At the beginning, this version was mainly used for the support of individual work and thus the majority of requirements targeted additional support for individual work.

The second version refined existing functionality and added new functions for a user-specific document processing.

The main requirement for the third version was the integration of the POLITeam system with the organization specific IT-infrastructure. This abolished the situation of the POLITeam users as being on an "island" within the organization. To enable the use of the POLITeam system along hierarchical lines, new functions were included that focused on the specific requirements of managerial users.

As described in the next section, in the course of time the users' understanding of groupware increased and more mature group work occurred based on the growing congruency of individual work patterns. This led to more elaborated requirements for a smoother cooperation and means for an informal coordination of work. e.g. for the support of conventions in shared workspaces, informal substitution regulations, and cooperative use of institutional mail inboxes. These needs were addressed in the final version.

POLITEAM USE

In this section, we present empirical results showing the use of the system over three years which points to different phases in groupware learning leading to congruent usage.

Method

The method used for identifying the phases in groupware learning was in-depth field research, supplemented with user interviews and protocols from five design team-user workshops. User interviews were conducted at three points during the project: before the system was installed, in order to determine work requirements, midway through the project, and at the close of the project after three years.

The site visits were documented in logbooks. The user advocates recorded where users had problems in system use and user requirements, along with some general observations. In order to track the process by which group members merge their individual work patterns into congruent system usage, user experiences were categorized into three categories using content analysis. The coding was checked by a second person. Where a few discrepancies were found, they were discussed and recategorized. The categories were:

- *General System Understanding:* Problems that concerned understanding single-user system functionality, e.g. Word, Excel, icon views, etc.
- *Individual Use Events:* Problems and experiences that concerned system handling of all aspects that do not concern shared objects, e.g. setting up individual document templates, organizing personal directories.
- *Group Use Events:* Problems and experiences that concerned system handling and understanding of shared objects.

Phases of system use

As a result of this categorization, rough phases of group development with the system were identified, and are shown in figure 1. The events in General System Understanding were so few that they were combined with Individual Use Events for the description. The six month divisions seemed to be a reasonable amount of time in which to characterize system use; smaller time categories had less data, and time frames longer than six months lowered the precision. We can describe these phases qualitatively as follows.

Initially, we see by far in the first six months that most of the group events concerned setting up group functionality, by the typist and Unit leader: shared folders, assigning colors for editing in Word, and new access rights profiles for common document templates. A few requirements emerged at this point concerning awareness: the typist wanted automatic notification to inform her clients when documents were completed, and the Unit leader wanted to know who made changes to a shared object. Also, the users discovered that they needed file codes for the common documents. In terms of individual events, here we see users setting up their own information structures. We also see a lot of "growing pains", i.e. problems with general system understanding: the Word menu, enlarging icons, formatting, etc.

In the second six months, we see more setting up of group functionality, mostly by the Unit leader: an address list, shared calendar, and internal shared folder. A few groupware requirements emerged: different access rights for different groups, and changing the date of the shared folder to reflect changes in the contents. Single-user events also concerned mainly general system understanding.

During the third six months, no new functionality has been set up, but it is now being refined. This concerns mainly the document template collection, since the users and typists are now established in a group work pattern and are now exchanging a variety of documents quite regularly. For example, the typist wanted access to a Unit member's document templates to ease her work. Single-user events show that users are beginning to customize features of the system, e.g. setting up personal document template collections, and creating email directories.

After one and a half years of system use, the group events are similar to the last phase: functionality is continuing to be refined in minor ways. A few requirements occur: the Unit leader wants to be able to establish shared templates for circulation folders, and employees should be identified in shared document printouts. Interestingly, two (new) users report that the naming conventions for documents is not clear to them. Single-user events also continue with minor understanding problems.

After two years of system use, we see a definite change to the events that are occurring. Many users are now reporting problems that they are having as a result of other users' actions: users are concerned that files are taken out of the shared folder and are lying on others' personal desktops, and there is confusion reported in how addresses in the shared address book are organized. First trials of electronic circulation folders were initiated by the newly-connected managers of the Units which led to the report of stalled folders (as with paper folders). Single-user events are minor, generally dealing with small problems with Word.

In the last six months of the project, again we see more problems reported that concern other users' actions: e.g. only owners can erase documents. The Berlin users want to identify only changes from specific users in Bonn, not all users. Also, Berlin and Bonn Unit leaders need specific colors assigned for editing. In addition, we see refinements for the awareness feature, e.g. reporting changes made on objects within a shared folder, and showing changes between Bonn and Berlin users. In addition, important requirements emerged concerning establishing a registry for documents. Single-user events continue as minor adjustment problems.

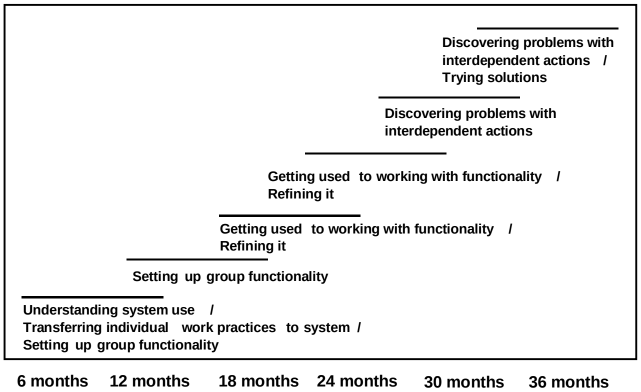
Discovering problems with interdependent actions / Trying solutions
Discovering problems with interdependent actions
Getting used to working with functionality / Refining it
Getting used to working with functionality / Refining it
Setting up group functionality
Understanding system use / Transferring individual work practices to system / Setting up group functionality
6 months
12 months
18 months
24 months
30 months
36 months

Figure 1. Approximate stages of groupware learning.

What is interesting for us, is that we see the group going through a process of developing congruent actions. We especially see a number of events that suggest that after about two years, now that group functionality has been set up, and work patterns seemingly firmly established, the group members now report problems that concern coordinating their work patterns. We interpret that at this point the group members are beginning to recognize the consequences of their interdependencies. We have some evidence that shows that the same types of problems with interdependencies occurred very early on in the system use, yet were not considered by users to be due to the actions of other members. For example, in the first three months of system use, the typist complained that an icon which usually turned blue was now turning black due to a system problem. Yet the typist did not make a connection that the color change was caused by a status change due to another user's action. This suggests that only with continued system use, the users gradually became aware of how others' actions were affecting their own system use, and they adjusted and accommodated their behaviors accordingly.

We see the workshops as playing an important role in facilitating users' recognition of how their work patterns interacted. In the first workshop, the users already began discussing basic conventions concerning naming, storing, and deletion of objects. In the second workshop, the users discussed categorization of shared documents through the assignment of file codes, as well as common structuring of the folders. In the third workshop, which took place after one and one half years, conventions for new situations were discussed, e.g. for cases of substitution, or when a member leaves the team. Also, conventions were elaborated concerning the appropriate technical means for a cooperative task, and general system use behaviors, e.g. starting the system in the morning, and checking mail regularly.

Although conventions were discussed heavily at the workshops, particularly the latter ones, the work to make the conventions work was actually delegated by the users to the design team, as opposed to being self-regulated. This was not our intent; our intention was to facilitate thinking about conventions, and we had expected that the users would take the responsibility. This was an indication to us that following conventions requires overhead and that technical support may be needed.

LESSONS LEARNED

At this point we have presented an overview of our experiences with the design approach and POLITeam use. During this process of designing a system in tight cooperation with users, we have learned a number of lessons with respect to group work practices, as well as on design issues for groupware. In the following, we have selected those that we feel best portray the challenges of continually integrating design with system use.

Lesson 1: Opening windows of opportunity

Research has shown that during the introduction of a groupware system, there appears to exist a limited timeframe in which modifications to a system can opportunistically be made [26]. Beyond this frame, changes are difficult to introduce since work patterns with the system become routinized, making system changes harder to be accepted.

In our view, windows of opportunity refer not only to system modifications but also to enabling the development of the appropriate group dynamics involved in groupware use. Group work requires a continual adaptation process to changing situations and tasks. This requires that group members are willing to continually reflect on their working habits, and on what is needed for their work. Windows of opportunity must therefore exist and be kept open (or reopened) to insure flexible adaptations of work patterns. Otherwise, the willingness for adaptation decreases, as does the potential for improvements of individual usage and cooperative processes. However, our experience has shown this window to be influenced by specific events, as well as the course of time.

The workshops became a forum in which events occurred to reopen windows; in fact the users themselves took the initiative. Although the designers intended the workshops for discussing design issues, the users took over and discussed their individual and group work contexts and practices. For example, a shared folder containing public information, which had been neglected, increased in use after the users discussed it (although curiously, the discussion concerned defending why it was not used). The user advocates also served to keep the window for change open. When problems with the integration of the house email system occurred (stage 3 of the system development), the user advocates stimulated the usage of the email by raising awareness of the external email functionality after the problem was solved.

Lesson 2: Lengthening the acceptance and tolerance window through user trust and resilience

Long term cooperation requires a trustful relationship between users and the design team. We have learned that the foundations for that relationship are laid in the first phase after system introduction and that serious consideration and immediate reactions to the user requirements are factors for building this relationship.

In the initial interviews, the impression of the design team was that the users viewed the introduction of the POLITeam system with a mixture of expectation and skepticism, motivation and anxiety. Common reactions from users were that they would need a lot of teaching, and in fact, they requested it. To achieve a trustful relationship and increase acceptance of the system, the user advocates stayed full-time for two weeks in the users' working environment. In this way, the advocates could immediately spot new requirements and feed them directly back to the design team in order to find a quick solution that could be realized on-site. For example, urgent requirements such as a reconfiguration of access rights, interface terminology, and document templates could be fulfilled directly on-site by adaptations and configurations of the system platform. For more complex problems, e.g. awareness, workspace administration, and integration of a registry, the designers came up with proposals for solutions which were discussed with the users at workshops or at site visits. The Ministry location in Bonn was closely located to the designers' site. In case of technical problems, site visits were possible within one hour.

Later during the project, users often mentioned that the immediate reaction to their needs contributed to the feeling that they were taken seriously as an important factor in the design process; they were not just treated as guinea pigs. By experiencing that their requirements were taken up and realized in new system versions, users became more tolerant to system functionality that was insufficient or lacking. They believed that these issues would be solved in the future and continued the system use instead of becoming frustrated. This relationship was later put to a test when the installation of a new version led to a dramatic setback of system performance at the client-PCs. Although users complained, system use did not drop since they expected that soon a solution would be found based on their previous experiences. However after the problem was solved by increasing the system memory in each PC, the users made it clear that again the fast reaction was important.

This observation adds a new dimension to the "windows of opportunity". Whereas the technologies studied in Tyre and Orlikowski [26] already passed the test of technical and organizational feasibility, POLITeam actually was the test for the technical and organizational feasibility of a groupware system in a ministerial setting. It was crucial to keep the tolerance window open to make the technology work for the users, and to do this, it was necessary to create a situation in which users felt that their problems and requirements were taken seriously and fulfilled within a foreseeable time-span. Our experience thus shows that the events that keep the window of acceptance open are immediate feedback on users' needs and a quick inclusion of urgent requirements in new system versions.

Lesson 3: Users cannot always predict their own system needs

Only after some time of system use, could the users gain a deeper understanding of how groupware could support their work. We discovered that what users may judge as attractive for their work practice at the outset may not be what they actually use. One user expressed this as, "Now we know how we work". In all of our discussions with potential users, the electronic circulation folders were judged to be the most attractive feature, even when they were tried during a laboratory workshop. Although improved during the course of the project, electronic circulation folders turned out not to be suitable for the particular tasks of the users. The users could not have predicted that: 1) they would use workspaces as the means for the permanent organization of their cooperative work; and 2) joint text production would be done in the shared workspace, instead of email exchange as the typists used before the groupware introduction. Not using the electronic circulation folders may be due to the horizontal cooperation of the users; a different finding may occur when hierarchical cooperation exists. At the end of the project, it was occasionally used in the cooperation with the Unit leader's superior.

Another finding for us was that something that appears theoretically reasonable for users may turn out quite different in practice. For example, the users requested that the system should ask for a file code upon document creation. However in use, most users did not obey the command, specifying a dummy code, and some complained that they felt "dominated by the system". This made it especially clear to us for the need to carefully understand the work process underlying system use [4].

Lesson 4: Supporting users in real work provides better feedback for system development

By gathering user requirements in the course of real work, we were able to learn many needs that were not articulated during interviews; this is due to both observing and supporting users in their situated work environment [23, 25]. The user advocates were external to the group; for this reason they were able to discover discrepancies among group members in their real work. They could develop an overview, seeing the whole cooperative network which the users themselves had trouble seeing. For example, it was discovered that different user groups adopted different naming conventions for the same documents. Also, some users worked on shared documents inside the shared workspaces whereas other users pulled the shared documents onto their personal desktops. The writing office manager complained that she no longer had access to the documents when she needed them. Removing documents from the shared workspace reduced the transparency and visibility of the cooperation.

Yet the user advocates also changed the way that users worked by presenting others' viewpoints, mediating, and discussing with users alternatives or limitations of the system. This support provided high quality requirements, e.g. membership administration, awareness, and conventions, since they were expressed in a situated context, often immediately, using the POLITeam system as the expressive medium.

Lesson 5: Conventions are eased with technical support

We learned that for a group to become congruent in its groupware behavior, a variety of conventions are needed. One can differentiate between conventions which can only be enacted through social means, and others that need technical support. For example, conventions that require extra actions on particular objects can be ideally supported technically by flexibly attaching configurable enablers that perform these actions automatically [24]. In POLITeam, a convention was established that removal of an object from a folder should leave an alias in the folder. This was supported technically with the option of applying it to particular folders. By applying a technical implementation that supported the user in applying the convention without enforcing it, it was easier for the users to follow the convention of not removing objects from the shared folder. Our lesson in this case is that designers need to give special consideration to conventions that require extra cognitive or manual overhead by the users.

Lesson 6: A medium approach requires awareness

The medium approach enabled a self-organization of cooperative work within the group instead of prescribing a sequence or turn-taking of cooperative actions. We learned that it requires the provision of awareness features to make the cooperation transparent and to avoid additional overhead for the coordination of situated activities. In the first system version, we excluded all functions that had the potential of being misused for purposes of controlling other users' system behavior, e.g. notification of changes of documents. We had the fear of getting in trouble with the union and many users during the initial interviews mentioned their concern about being controlled. However, the extensive use of shared folders led to a number of requirements for a notification functionality for the actions of users on shared objects. Users were tired of sending notification emails to inform others who were dependent on their results. Thus, it was necessary to provide notifications to make cooperative actions visible and to inform partners about intermediate actions or the completion of a task.

The user advocates discovered that awareness notifications were useful for understanding the system behavior in asynchronous work processes. In particular, it can aid users in attributing system behavior to another user's action, thus facilitating the users in thinking of a cooperative system use model, as opposed to a model for individual system use. Before awareness facilities were introduced, one typist misunderstood the disappearance of a document from a folder not as a consequence of another user's action, but as a failure of the system.

It became clear that the awareness mechanisms need to be tailorable to different user groups, different contexts, and individual preferences. For the Bonn users, identifications of all authors of text modifications were relevant. However, what was not foreseen was that for Berlin users, only the final changes made by the two Bonn leaders would be relevant. Seeing all changes was confusing and created overhead for the Berlin users.

Lesson 7: Prototypes are design specifications

Prototypes are a valuable tool for cooperative prototyping studies [2] and as a "proof of concept". We have learned that during the design process they serve also as the design specification. In the early project phase, many prototypes have been developed to evaluate the practicality or realizability of the different requirements. These initially quite simple prototypes were quickly extended and revised in several versions, as more requirements and design proposals and ideas were born. However, many versions were abandoned and thrown away by a frustrated designer after it was evaluated within the design team, until we reached a stable version that was released for a new system version at the users' site. We often asked ourselves if we could have done a better job, by a more detailed system specification in advance in contrast to quickly implementing ideas in a prototype. However, our lesson learned that groupware cannot be evaluated prospectively on a theoretical basis answers that question. In most discussions concerning a design decision, the design group was fairly confident that we would have found a good solution. But when experimenting with the resulting prototype, immediately new perspectives on the problem, as well as the interplay of the new component with other functionalities were revealed. The different perspectives of the designers and the users, represented through the user advocates, allow only an iterative approach to the final system design. Our experiences show that instead of a theoretical paper-based design specification, it is the prototype that plays the role of the design specification in the inner cycle of our cooperative design approach.

Lesson 8: The potential and limitation of off-the-shelf groupware platforms

An off-the-shelf groupware platform proved to be a good basis for a cooperative design in work practice, but it needs to fulfill a number of requirements to enable an adaptation to the work setting and to allow sophisticated developments.

The LinkWorks groupware platform provided the basic tools for the realization of the POLITeam system components. It offers an object-oriented API (Application Programmers Interface) for a functional extension and integration of external applications. Our experiences with this platform led us to the following general requirements:

- The integration of external applications is important for embedding the groupware platform into the existing IT-infrastructure.
- The platform must support an adaptation to the specific terminology of the work setting to support the mapping of real-world artifacts to their electronic counterparts.
- The initial introduction of POLITeam required a reduction of the overall functionality: to reduce complexity, to exclude functionality that could be misused for controlling purposes, and to support the building of shared usage and cooperation patterns.
- Prototyping tools should be provided either by the platform itself, e.g. through scripting languages, or must be capable of being integrated.
- The programming interface should not only provide methods for an extension of the functionality, but also for a modification of the behavior of basic functionality, i.e. access control mechanisms, listing

events of user actions, presentation methods at the graphical user interface, and system consistency [6].

Although these experiences are based on a particular platform, we believe that they are general enough to be applicable to other platforms too. In fulfilling these requirements, a groupware platform possesses a high potential for cooperative design in work practice, since it frees the design team from investing too much effort in building and dealing with the technological infrastructure. It enables concentration on the fine tuning of specific aspects of cooperation support. However, the limitation comes to pass, when the last requirement is not accomplished. In our case the realization of awareness features [22] and a group-based tailoring functionality [24], was neither supported on the client nor server side of the platform. As a result, the realization of these functionalities in a new client made a reimplementation of the standard groupware-client necessary which required 50%-60% of the whole development effort. Unfortunately, it is almost impossible to anticipate these development problems, since a user-driven approach makes it difficult to foresee upcoming user requirements.

SUMMARY AND CONCLUSIONS

In this paper we presented our experience from four years of system development including three years of users' work practice. Our goal was to support group work with appropriate technology, within the constraints of a ministerial setting. We discovered this to be a continual and dynamically changing development process. What contributed to these dynamics were the interactions and influences between the system development, design process, and system use. Interestingly, the rates of change in these components were not constant, but rather fluctuated over the course of the project. Sometimes we observed many effects occurring rapidly, e.g. an increase in requirements after a new POLITeam version was introduced, or increased system usage after a workshop. In these cases, the events that opened the "windows of opportunity" (e.g. workshop discussions) were valuable because they lengthened the timeframe in which system features could be accepted, and they increased the openness of the users towards experimentation.

Both the design process and system design must support the congruency process of a group. This involves observing whether users have the same assumptions about their cooperative work with the system, and helping users to become aware of, and to find methods to coordinate their interdependencies. Interdependencies may arise through planned cooperation and use of the functionality, but there also exist interdependencies in a group of users who are simply using the same shared objects without explicit intent to cooperate with these objects. Experiencing and understanding the implicit dependencies of groupware use requires learning over time as well, as does the use of groupware features for intentional cooperation. An important factor for achieving this is the technical support of awareness that can make people's actions and the group context visible, and therefore facilitates the management of interdependencies. User conventions appear to be necessary to adjust different work patterns among a group. However, obeying conventions may involve enough extra effort so that they are not followed. Technological solutions can ease the following of conventions, yet the challenge is that they remain flexible to adapt to the local contingencies of situated work.

The design process should function to keep windows of opportunity open so that users are open to new features of the system. Routinized system use can lead to the danger of preventing users from being open to change, particularly change which may benefit them. With the media approach we used, system use is not very predictable. Therefore, it is especially important that users do not become routinized too quickly in their system usage, which would hinder the potential for emergent change in their work patterns. The design approach must also react flexibly to emerging collaboration patterns.

Perhaps one of the strongest lessons that we have learned in the course of the project is that the design process itself must take the situated nature of group work into account. The well-elaborated four-year project plan used for funding was soon relegated backstage as we became aware that we needed to adapt our methods to new situations in the group's work as they arose. Formal business process analyzing of workflows, formal object-oriented specification of system functionality, and formal project plan milestones were abandoned, and instead scenario-based understanding of situated work processes, rapid prototyping, and user-driven priorities were adopted.

Although we hope that the POLITeam system will remain in the user organization after the project is over, it is not yet clear. At the time of this writing, the IT-department is still in the decision phase whether or not POLITeam becomes the organization-wide strategic groupware system. Discussion of the worst case scenario with the pilot users reveals that although the technical system may be withdrawn, the ideas and concepts will survive. The pilot users will look for ways in how to apply the cooperation patterns they developed in POLITeam to the standard organization-wide system, which supports simple file sharing and email. Furthermore, the IT-department involves some of the pilot users in the decision process. We were glad to hear that all users are willing to participate again in such a pilot test. The reasons for this are that they felt that they became a valuable partner who influenced and determined the design process, and that they have achieved an increased competence in technical, organizational, and social aspects of groupware. It was not just us who learned a list of lessons, but also the users as well.

ACKNOWLEDGMENTS

Many thanks to Wolfgang Gräther and our anonymous reviewers for comments on this paper, to all other members of the POLITeam project and especially to all POLITeam users for their motivated participation in the project.

REFERENCES

1. Bentley, R. and P. Dourish. Medium versus mechanism: Supporting collaboration through customization, in *Proc. of Fourth European Conference on Computer Supported Cooperative Work (ECSCW '95)*, Stockholm, Sweden, H. Marmolin, Y. Sundblad, and K. Schmidt (eds.), Kluwer, 1995, 133-148.
2. Bødker, S. and K. Grønbæk. Cooperative Prototyping Studies - Users and Designers Envision A Dental Case Record System. In *Studies in Computer Supported Cooperative Work - Theory Practice and Design*, J.M. Bowers and S. Benford (eds.), North-Holland, Amsterdam, 1991, 315-332.
3. Bowers, J. The Work to Make the Network Work: Studying CSCW in Action, in *Proc. of Conference on Computer Supported Cooperative Work*, Chapel Hill, NC, USA, R. Furuta and C. Neuwirth (eds.), ACM Press, 1994, 287-298.
4. Button, G. and R. Harper. The relevance of "Work Practice" for Design, *Computer Supported Cooperative Work: The Journal of Collaborative Computing 4*, 4, 1996, 263-280.
5. Button, G. and W. Sharrock. The Production of Order and the Order of Production: Possibilities for Distributed Organisations, Work and Technology in the Print Industry, in *Proc. of Fifth European Conference on Computer Supported Cooperative Work*, J.A. Hughes, *et al.* (eds.), 1997, 1-16.
6. Dourish, P. The Parting of the Ways: Divergence, Data Management and Collaborative Work, in *Proc. of Fourth European Conference on Computer-Supported Cooperative Work: ECSCW '95*, Stockholm, H. Marmolin, Y. Sundblad, and K. Schmidt (eds.), Kluwer Academic Publishers, 1995, 215-230.
7. Grudin, J. Why CSCW Applications Fail: Problems in the Design and Evaluation of Organizational Interfaces, in *Proc. of CSCW ´88*, Portland, Oregon, ACM, 1988, 85-93.
8. Klöckner, K., P. Mambrey, M. Sohlenkamp, 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, in *Proc. of ECSCW '95*, Stockholm, Kluwer, 1995, 17-32.
9. Kyng, M. Experience with Participative Application Development, in *Proc. of IFIP 13th World Congress*, K. Brunnstein and E. Raubold (eds.), Elsevier Science B.V. (North Holland), 1994, 107-119.
10. Malone, T.W. and K. Crowston. What is Coordination Theory and How Can it Help Design Cooperative Work Systems, in *Proc. of CSCW ´90*, Los Angeles, USA, ACM Press, 1990, 357-370.
11. Mambrey, P. and A. Tepper. Metaphors and System Design. In *Computers As Assistants. A New Generation of Support Systems*, P. Hoschka (eds.), Lawrence Erlbaum Assoc., Mahwah, New Jersey, 1996, 269-280.
12. Mark, G., L. Fuchs and M. Sohlenkamp. Supporting groupware conventions through contextual awareness, in *Proc. of Fifth European Conference on Computer Supported Cooperative Work*, Lancaster, U.K., J. Hughes, *et al.* (eds.), Kluwer Academic Publishers, 1997, 253-268.
13. Moreland, R.L. and J.M. Levine. Newcomers and Oldtimers in Small Groups. In *Psychology of Group Influence*, P. Paulus (eds.), Lawrence Erlbaum, Hillsdale, NJ, 1989, 143-186.
14. Newcomb, T.M. Interpersonal Balance. In *Theory of Cognitive Consistency - A Sourcebook*, R.P. Abelson (eds.), McNally, Chicago, 1968, 10-51.
15. Orlikowski, W.J. Improvising Organizational Transformation Over Time: A Situated Change Perspective, *Information Systems Research 7*, 1, 1996, 63-92.
16. Paetau, M. Configurative Technology: Adaptation to Social Systems Dynamism. In *Adaptive User Support - Ergonomic Design of Manually and Automatically Adaptable Software*, R. Oppermann (eds.), Lawrence Erlbaum Associates, Hillsdale, 1994, .
17. Pankoke-Babatz, U. and A. Syri. Collaborative Workspaces for Time Deferred Electronic Cooperation, in *Proc. of GROUP'97: International ACM SIGGROUP Conference on Supporting Group Work*, Phoenix, AZ, S. Hayne and W. Prinz (eds.), ACM Press, 1997, 187-196.
18. Prinz, W. and S. Kolvenbach. Support for Ministerial Workflows, in *Proc. of Conference on Computer Supported Cooperative Work (CSCW'96)*, Boston, MA., M.S. Ackermann (eds.), ACM Press, 1996, 199-208.
19. Robinson, M. and L. Bannon. Questioning Representations, in *Proc. of ECSCW ´91*, Amsterdam, Netherlands, L. Bannon, M. Robinson, and K. Schmidt (eds.), Kluwer, 1991, 219-234.
20. Rogers, Y. Coordinating Computer Mediated Work, *Computer Supported Cooperative Work: The Journal of Collaborative Computing 1*, 4, 1993, 295-315.
21. Rogers, Y. Exploring Obstacles: Integrating CSCW in evolving organizations, in *Proc. of Conference on Computer Supported Cooperative Work*, Chapel Hill, NC, USA, R. Furuta and C. Neuwirth (eds.), ACM Press, 1994, 67-77.

22. Sohlenkamp, M., L. Fuchs and A. Genau. Awareness and Cooperative Work: The POLITeam Approach, in *Proc. of HICCS'97*, Maui, Hawaii, IEEE Computer Society Press, 1997, 549-558.

23. Suchman, L.A. *Plans and situated actions. The problem of human-machine communication*, Cambridge University Press, Cambridge, 1987.

24. Syri, A. Tailoring Cooperation Support through Mediators, in *Proc. of ECSCW'97: Fifth European Conference on Computer Supported Cooperative Work*, Lancaster, UK, H. Hughes, *et al.* (eds.), Kluwer Academic Publishers, 1997, 157-173.

25. Twidale, M., D. Randall and R. Bentley. Situated Evaluation for Cooperative Systems, in *Proc. of CSCW'94: Conference on Computer Supported Cooperative Work*, Chapel Hill, NC, USA, R. Furuta and C. Neuwirth (eds.), ACM Press, 1994, 441-452.

26. Tyre, M.J. and W.J. Orlikowski. Windows of Opportunity: Temporal Patterns of Technological Adaptation in Organizations, *Organization Science 5*, 1, 1994, 98-118.

27. Weick, K.E. *The Social Psychology of Organizing*, Addison-Wesley, Reading, MA, 1969.

28. Williams, J.S., J.D. Martin and L.N. Gray. Norm formation or conditioning, *Small Group Behaviour 6*, 1975, 141-150.