Developing Groupware with Evolution and Participation A Case Study

Helge Kahler
Institute for Computer Science III
University of Bonn
Roemerstrasse 164
D-53117 Bonn
+49 228 73 4276
kahler@cs.uni-bonn.de

ABSTRACT
This paper is about experiences with the evolutionary and participatory development of a search tool for a groupware system. After the description of different software engineering approaches and their use for evolutionary and participatory software development the POLITeam groupware project is presented. The procedure of how the search tool for POLITeam was developed including interviews, workshops and the usage and evaluation of prototypes is described. The resulting search tool is presented. The paper concludes with remarks about the usage of participatory design methods for the introduction and customization of generic groupware in different organizational settings.

Keywords
Participatory development, groupware, electronic search in groups, prototyping, customization, introduction of generic groupware

INTRODUCTION
Approaches to Software Design
For a long time the development of software applications was mainly technically determined. The top-down waterfall model of the software life cycle (cf. Boehm 1976) and revised versions of it became the standard for software development. While this model proved to be appropriate for some classes of software, it didn't work well with others. Particularly for the development of "embedded programs" (Lehman and Belady 1985) that are characterized by the interdependence between the software and its environment the waterfall model proved to be inadequate. Several software engineering approaches and software life cycle models have been developed to overcome these shortcomings that give more consideration to the organizational environment of the program-to-be. Among those are Boehm's spiral model, Henderson-Sellers' object-oriented fountain model, Hesse's EOS model, and Floyd's STEPS model.

In *PDC'96 Proceedings of the Participatory Design Conference.* J. Blomberg, F. Kensing, and E.A. Dykstra-Erickson (Eds.). Cambridge, MA USA, 13-15 November 1996. Computer Professionals for Social Responsibility, P.O. Box 717, Palo Alto CA 94302-0717 USA, cpsr@cpsr.org.

In Boehm's risk-driven spiral model (Boehm 1988) several cycles are involved each of which includes the planning of the next phase, determining objectives and constraints, evaluating alternatives and resolving risks, and developing the next-level product. With the emergence of object-oriented programming, analysis and design Henderson-Sellers and Edwards (1990) proposed their fountain model for the object oriented life cycle. It is based on the iteration and overlapping of consecutive phases (e. g system design, program design and coding) and on overcoming the need to freeze specification at an early stage by using autonomous classes that can easily be modified without having strong side effects on other parts of the system. Another approach involving object orientation is Hesse's EOS model (Hesse & Weltz 1994). It is based on merging evolutionary system development with the principles of object orientation. Analysis, design, implementation and application are considered to be the four activities of a software development cycle that are performed on the system-, component-, and class-level with increasing frequency. The EOS model is explicitly based on the idea that software projects create technical artifacts while shaping the structure of work in a particular organization, thus dismissing the notion of software development as a mere engineering process.

All of these approaches stress the importance of the organizational environment for software development with the overall notion that the design of software should be worked on beyond the early stages of a software's life cycle but must contain evolutionary aspects that allow for design changes and adaptations during software development.

Floyd's STEPS model of software development (Floyd et al. 1989a) explicitly introduces a new aspect into software development for embedded programs. It is strongly inspired by the Scandinavian approach (cf. e.g. Floyd et al. 1989b, Greenbaum & Kyng 1991, Ehn 1993) to system design with its stress on user participation (also cf. Floyd 1993). Incorporating strong user participation STEPS bridges the gap between software engineering and the discussions about participative software design lead in the Participatory Design (PD) and Human-Computer-Interaction (HCI) communities. STEPS is meant to develop embedded programs

not only in an evolutionary process but with users playing a decisive role in the development process. Software development is seen as a process of mutual learning where the developers contribute their knowledge of formal methods and software development and the users contribute their knowledge of the work domain. In the STEPS model each of them have tasks in the development process with some of the tasks being common (see Figure 1).

Developing Groupware

The question of how to develop software that is strongly embedded in the organizational environment is particularly important for CSCW (Computer Supported Cooperative Work) research. Here, a group's particular ways of communicating and cooperating need to be supported. These can be vastly different between different groups and might also change within one group in the course of time. In order to be able to develop adequate software to support such a group it is necessary to find out the group's needs and then develop or adjust the software accordingly. This should be done in a process that includes both participation and evolution. Participation of members of the work group gives them the chance to put in their work and group experiences while evolutionary development of the software is necessary since it is hardly possible to meet the software needs of a dynamic system like a work group with a software right away and without adjusting the software along with the experiences made in the work group.

Although it was originally not made up for the development of groupware the intriguing aspect about the STEPS model is that it combines user participation in different parts of the process the with a cyclic approach allowing for stepwise improvements of the existing prototype or program version. Thus, the particular difficulties of developing an embedded program can be faced in an appropriate way. User participation in the design phase helps to understand the structure of work and the particular needs of an organization or a group of users while the cyclic evolution of the program is bridging the gap between specification and usage by having the software gradually approximate to the current work practice. Considering the growing environmental dynamics and complexity organizations have to deal with and the emergence of post-tayloristic forms of organization more and more programs will be strongly embedded in organizational settings and will need to be developed accordingly. Some authors have remarked that STEPS has only little focus on the actual participatory activity and does not involve exploratory prototyping (Grønbæk et al. 1995). While this is true as far as explicit statements go, STEPS provides a good base to work on and needs to be filled with concrete actions when working on system development.

So, being based on the idea that software development should be an evolutionary and participatory process the STEPS model can be considered to be a good start for evolutionary and participative development of groupware with all its special aspects to be taken into account. This is why it was decided to use STEPS in the POLITeam project.

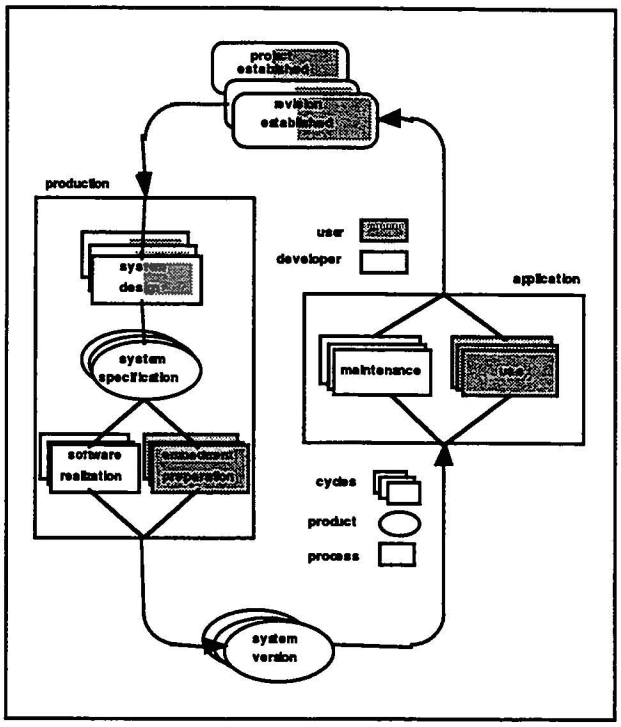

Figure 1: STEPS model for software development (Floyd et al. 1989)

The POLITeam Project

In 1989 the wall between East and West Germany came down. This resulted in many social, economic, and political changes one of which was the decision that Germany's capital was to move from Bonn to Berlin. Since the movement of such a big administrative organization with some thousand employees could only be done stepwise and since it was decided that some of the German federal administration was to remain in Bonn the government faced the need to come up with ideas to support the now geographically distributed government. Different parts of the government that were only miles apart in Bonn were to be partly in Bonn and partly in Berlin with a distance of about 400 miles. Among other activities the government set up the POLIKOM program to support research and development of adequate ways and tools for telecooperative work.

Taking part in this program is the POLITeam project consisting of industrial partners (VW-Gedas as software company), research institutes (University of Bonn and GMD, the German National Research Center for Information Technology) and application partners from the federal administration, a state administration and the software engineering department of a car manufacturer. The aim of the POLITeam project is to develop a system to support distributed work in large organizations. This is done by providing a workflow component to handle circulation folders that structure the workflow and by implementing the metaphor of a "shared desk" that integrates document processing tools. This means that the users of the POLITeam application work on a desktop where they can place objects that others have access to, e.g.

shared folders or text objects that are editable by a group of persons (cf. Klöckner et al. 1995).

POLITeam is based on Digital's LinkWorks™. The functionality of LinkWorks™ is used, enhanced, and changed by adding software components and using the LinkWorks™ application programming interface. POLITeam is a client/server application where usually each client provides document processing applications (e. g. Word for Windows) while the server stores the documents and meta-information like access rights, a list of persons who are to receive a circulation folder, or the position of objects on one's desktop. The design approach of POLITeam explicitly emphasizes evolutionary and participative aspects and is based on Floyd's STEPS. For each of the application partners that were to introduce POLITeam into their organization their work and organizational structure was analyzed. After configuring the first versions of POLITeam to each of the application partner's needs it was introduced in their organizations so that about 40 persons altogether work with the system right now. In the course of the project more users will be provided with POLITeam. The introduction was accompanied by training the users to work with the system and after that the application partners were visited regularly by user advocates (cf. Mambrey et al. 1996), i. e. every week or fortnight, to give feedback about their experiences with the system and to suggest improvements for the upcoming next version of POLITeam. Learning from these visits and workshops that were held with the application partners the current POLITeam version will be reshaped to better meet the application partner's needs.

The following chapter provides an example of how user involvement resulted in system evolution for a tool from the POLITeam system.

DEVELOPING A SEARCH TOOL - EXPERIENCES

Existing Search Tool

The basic version of LinkWorks™ had a tool implemented that allowed for searching objects. With this search tool one could basically find any object known to the system. The search tool provided different search criteria for an object such as the name, the object class (e.g. "text" or "folder"), the date of its last change, the name of its creator and more. To protect the privacy of the workgroup's members the possibilities of the search tool had to be restricted by providing objects with a search flag that marks if an object can be found by the search tool. This flag cannot be set directly by the creator of an object but only via an access profile containing the information that this object is unsearchable.

With the application partners we agreed on three different access profiles that should be configured and provided for them with the option of refining the access profiles later (e.g. by allowing or prohibiting the attachment of an object to an e-mail) and thus increase the number of access profiles. The most general of the three initial access profiles for an object was "public" where every person is allowed to see / read and change / write the object. The second access profile was "for your information" meaning that the object could only be read but not changed by anyone but the creator and the most restrictive access profile was "private" where no one but the creator of an object could read or write it. Of these three access profiles "private" was the only one where the search flag of the object was not set so that this object was unsearchable, i.e. not visible for the search tool. By allowing for granting the "private" access right to objects and thus preventing them from being found by the search tool basic issues of privacy were ensured.

Still the search tool was expected to make problems in the daily work of the application partners so it was finally decided not to use the existing search tool at the application partners' sites but to develop a new search tool that should be more adequate to the users' needs. To understand the problems that arose with the original search tool some more of its functionality must be explained.

To support cooperative work on a document (e. g. text) LinkWorks™ provides three possibilities. The users can either work on one electronic document that is treated like a real world paper document. In this case there is only one copy of the document that can be worked on by one person at a time and that has to be moved to and fro for different persons to see or change it. The second way for cooperation is to make one or more copies of an existing document that are treated like real world copies, i.e. that can be worked on independently. If the aim is to produce a single document of these copies they must be merged manually. The third possibility is encouraged by LinkWorks™ and provides a way of handling a document that exceeds the possibilities of a paper document. Here, the document is shared between different persons in a way that they can all see this document on their desk at the same time. This is done by providing links from their desks to the document. If one person changes the document the links to the desks of others are immediately updated so they can see the changes. The advantage of sharing a document this way is that it is not necessary to send a document around for somebody else to change it or to send copies of a document around for others to be informed about the current state of the document. Moreover, working with links is more efficient than sending around copies that are worked on by different persons and that need to be merged afterwards.

Whenever the search tool was started it searched for objects in the system for that the specified criteria applied. So, if person A had created a text with the access profile "public" or "for your information" called "letter to J. Johnson" with a word processor and stored it in a folder on her LinkWorks™ desk then person B would find the text with the search tool request looking for all objects having the word "letter" in their name. Then the search tool would automatically create a link to this text and put it on B's desk in the "search" folder. The automatic creation of links by the search tool resulted in various problems concerning privacy aspects and data handling.

One problem consisted in the fact that person A was not informed about the fact that somebody searched her desk for an object and actually found one. Users working at the application partners' sites realizing that someone could

"snoop" on their desk which they considered a more or less private area they could feel uncomfortable about this. On the other side there is the need to search for objects in the system to get the information necessary to do the work. Moreover, for users it is extremely impractical to protect "their" objects from being found by giving them the access profile "private" since this would hamper shared editing of documents and cooperation in general.

Another problem caused the unintended deletion of files and was a major reason to decide for the redevelopment of the search tool. This unintended deletion resulted from the slightly inconsistent handling of files in the search window. The reason for this was that in the search window all objects found were represented as links to the original objects as described above. While in an "ordinary" window every deletion had to be confirmed, if someone pressed the delete key in the search window e.g. on a text found only the link in the search window was deleted without confirmation of the deletion and the object icon was removed from the search window but the original object still existed e. g. on someone else's electronic desk. The same was true for found and deleted folders. This folder could contain linked and unlinked objects. The impression the users could get was that any deletion of an object started from the search window was harmless since only the link would be removed. This, unfortunately, was not true since when users opened the found and thus linked folder it contained objects that were not necessarily linked themselves. So when they would delete an unlinked object, say a text, in the found and linked folder it would be deleted for all other users that had this folder on their desk. This could lead to an unintended deletion of unlinked objects contained in a found folder.

A third incentive to work on the search tool was that the initial phase of internal use of the search tool made clear that the abundance of search criteria made it difficult to use the search tool. This resulted from the fact that the developers had implemented all criteria that could technically be searched rather than restricting the search criteria to a useful subset.

Redevelopment of Search Tool

The experiences from the initial phase of internal use concerning the search tool strongly implied that the search tool had to be redesigned and reimplemented in order to solve the existing problems with it. While so far the search tool had been just one of many features of POLITeam the experiences of the users had made it one of a few special things and problems to focus on.

In order to develop a search tool that supported the work for the application partners adequately the shortcomings of the existing search tool had to be overcome. We considered the aspects of searching that have to do with the particularities of group work to be of particular importance. So we decided to not only find work-arounds to deal with what had proved to be solved badly with the existing search tool but to go deeper and find out more about searching in a group and about the conflicts coming along with it. Our goal was to develop an improved search tool and learn more about potential conflicts and possible solutions that are relevant for people working with a groupware.

In the course of the redevelopment of the search tool different techniques of user participation and software evolution were involved. We conducted 10 interviews with interview partners from four application partner organizations, held four workshops where aspects of searching were raised, two of which were dedicated to search tool prototypes, and we developed three prototypes of search tools which were later evaluated.

These techniques were meant to bring up different aspects of requirements for the search tool and can be considered to be concretizations of the user-related activities in software engineering models involving user participation.

Interviews

To get a better understanding of how search in a work group is performed we started with conducting interviews about how people who cowork with each other search objects, i.e. documents, papers, or folders in an office environment. We talked to ten people, two of which worked in a library, two in a state administration which is an application partner, three in the office of a software company, and three in the office of a construction company. We deliberately chose interview partners that had worked and others that had not worked with POLITeam to get input from a wide range of work practise and not be biased by users' previous experiences with POLITeam. The interviews were led with one person at a time, lasted about 30-45 minutes each and were conducted along a questionnaire with 29 questions that served as a guide which left space for additional questions and talk. The questionnaire consisted of open questions (answers in sentences, not just yes or no required), included physical and electronic search, and had two parts, the first of which related to the search activities of the interviewees in their offices (What are causes for a search? Describe how you go along? What tools do you use: telephone, post-it-notes etc.?), while the second related to privacy issues. Here, the interviewees were to take the roles of both a person searching something in a work group and person "being searched on", i. e. someone, who was asked about an object ("Do you know where this document is?") or whose room or desk or hard disk was searched by someone else (cf. Krüdenscheidt 1996). A similar role-oriented technique was used by Wulf & Hartmann (1994) researching on effects on visibility in a network.

The answers of the interviewees shed a light on different aspects of searching in a work group. Usually one of two problems is the starting point for a search, it is either the problem to find an object whose existence is known or the question if there is an object that contains the information searched. Three main causes for a search could be identified. These are the intention to work on a searched object (e. g. use components of an existing document to create a new one), the intention to gain information, and the intention to control something, e. g. the current state of a project, or someone. The objects searched were mainly internal (e. g. prepared speech for minister or inventory list) or external (e. g. legislative texts or offers from providers) text objects.

The ways how and where objects are stored in a particular work place differed in the different organizations. This includes organizational as well as personal storage. Several personal preferences could be found which the interviewees stated to be efficient for themselves. On the organizational level we found different structures to sort and order documents like order by date, by internal or external order numbers or by task areas and within them again by project number and date. Moreover, in each of the four organizations a central place for the collection of documents exists, e. g. a registry in the state administration. The organizational search was often started by limiting the time range of the object to be found and by providing key words or restricting thematic areas if the document order structure supported this search. Interviewees in three of the four organizations worked on a computer and searched with the Microsoft Windows file manager or the word processor file manger. Here, the predominant search criteria are the file name, date, key words, and the author of a document.

The interviewees stated that they involved others in their search when they needed help, e. g. from a person in the registry who knew "their" files or from a colleague who had worked with them on the document searched. Usually the others were not involved in the search process itself but by communication, i. e. they were contacted personally or on the telephone and asked questions about a document. For a search where others are affected the interviews showed a potential for conflict. The persons interviewed stated that usually the doors of their offices were open and that basically everyone could search in everybody else's room but that usually one wouldn't search in someone else's drawer but only on the desk and that this also depended on the relation of the persons. Potential conflicts showed where electronic search was discussed. Here, the symmetric design of the questionnaire allowed for every interviewee to take the role of a "searcher" and the role of a person "being searched on". In the role of a person searching actively the interviewees pleaded for a nearly unlimited access for electronic search arguing that this would be helpful and necessary for cooperation and adequate for team work. When they took the role of a person affected by someone else's electronic search they felt uncomfortable knowing that everyone could look into their folders and considered this as an unwanted intrusion. One person (working with another system than POLITeam) described her work practice where she would not move a document she worked on from her home directory that only she could access to a public directory until her work on the document was completed.

The interviews helped us to a deeper understanding of how people involved in team work search objects and they made clear that there was a particular need to handle the conflicts that might result from a search performed with a search tool on other person's electronic desks within POLITeam.

Workshops Related to Group Work

Besides the interviews in this first step of the redevelopment of the search tool two workshops were held with eight users of the federal administration (ministry) application partner where searching was discussed among other topics. We incorporated workshops with a group of users in the development process since we felt they could bring out much more of the group dynamics than the interviews were able to.

At this time they had used POLITeam for some while but they did not know the POLITeam search tool which had been disabled before the system was introduced there. In workshop I naming conventions for documents were discussed. The problem arose that in the office where documents were partly typed, processed and collected they used POLITeam and DOS without POLITeam and they were working with a very rigid name structure where document names had the DOS 8.3 form and where the first eight letters consisted of two letters for the document type (e. g. speech, letter, text from circulation folder) and the following six letters stood for the date. They did not want to change this rigid structure to stay compatible with the rest of the ministry. The people cooperatively working on the documents and writing the letters and speeches wanted to use POLITeam's facilities for long (32 letters) names without sticking to the rigid conventions. This showed that the individual representation of information was important and that POLITeam had to provide means to find objects that obeyed different naming or ordering criteria. The second workshop was held with the same group of users and served to introduce a new version of POLITeam where it was possible to order the contents of a folder by different criteria like name, date, or key word. Also a viewer for a fast preview of documents and a facility for tree-like hierarchical representation of objects in POLITeam were presented. The users said that these three features would be of great help in finding objects. While not being part of a special search tool they provide facilities to represent object names and other features in different ways that the users can choose between. Thus, individual preferences e. g. in sort orders and naming are supported. The tree-like hierarchical presentation as well as the possibility to determine the sort order are very helpful for location-based finding which widely used when working with user interfaces based on the desktop metaphor (cf. Barreau & Nardi 1995, Fertig et al. 1996). In the same workshop a search tool modified from the original search tool was introduced. This prototype 0 contained all the functionality of the original search tool except that a person could only perform a search on her own POLITeam desk which on one side meant that someone searching could not violate someone else's privacy because she could simply not access other electronic desks, but that on the other side cooperation and team work which POLITeam focuses on were extremely hampered. Moreover, the response time for the search results became very long since restriction to the desk of the initiator of the search made it necessary to first search all objects on her desk which included a time consuming check for every object in the system and then in a second search restrict the objects on the desk to those for which the search criteria applied.

Prototyping

After the interviews and workshop I and workshop II we felt we knew enough to program a prototypical search tool that was to incorporate what we had learned from the interviews

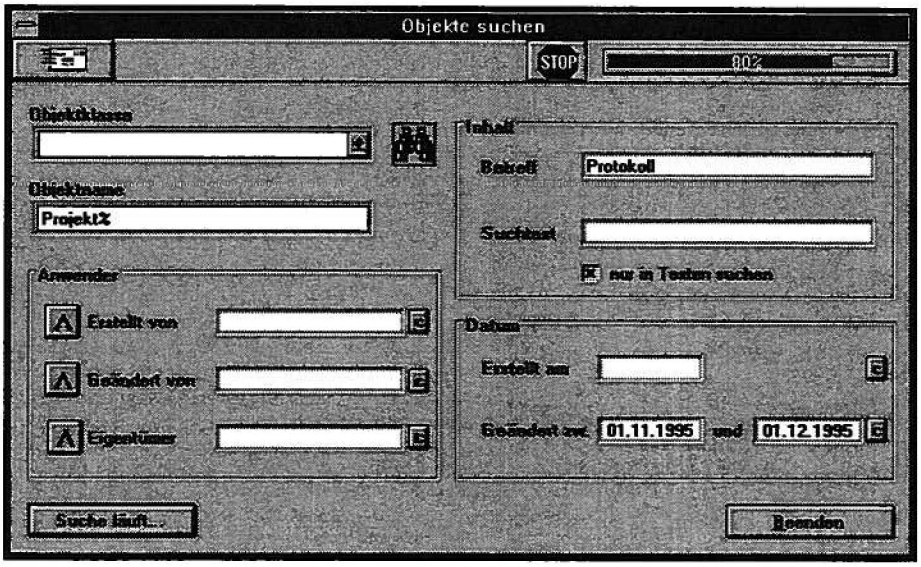

Fig. 2: Input dialog of search tool prototype.

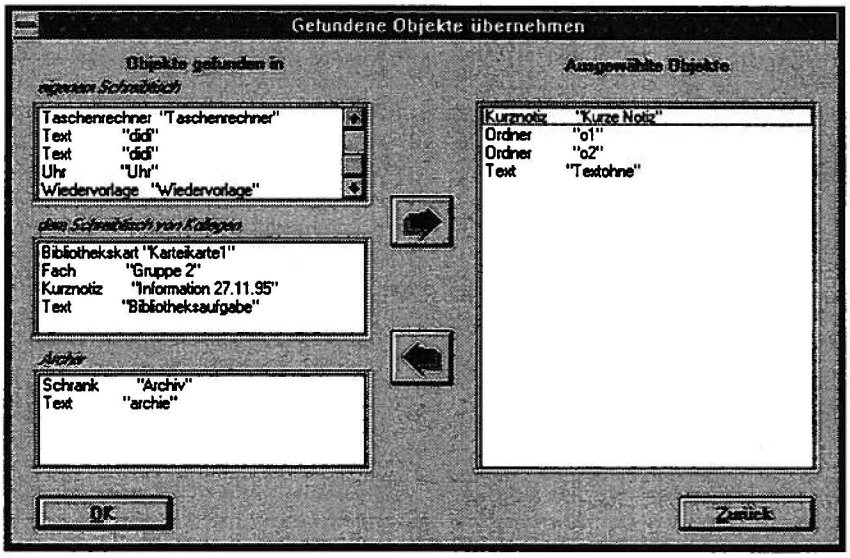

Fig. 3: Output dialog of search tool showing where objects were found.

and workshops. Two alternatives were to be considered. The first was to change the original search tool to fit the new requirements. While this would have had performance advantages the means of changing the original search tool provided by LinkWorks™ were not powerful enough to have us implement the features wanted. So in trading off performance for flexibility we decided to use an external programming language for the search tool prototype and access the LinkWorks™ objects by means of the LinkWorks™ programming interface.We choose Visual Basic as programming language and created a search tool that met the requirements in different ways. It included possibilities to search according to different criteria, among them the name and class of an object, the name of the person who created or owned or changed it, and the date or period when it was created or changed. It was also possible to search for a key word or search the complete object (usually a text object) for a text string. Moreover, to support the communicative aspect of the search, a button to activate the e-mail component of LinkWorks™ from the search dialog was implemented.

A major improvement was the distinction of the area where an object was found. For every object found it was indicated whether it was found on the searcher's own desk, on someone else's desk or in the archive of the group. Knowing this

the most interesting objects could be picked. For them a link was created in the search result window of POLITeam.

The indication for the found objects where they were found is a first step towards a conflict management necessary for a search tool for groupware and groupware in general (cf. Wulf 1995 for a general treatment of conflicts in groupware). Such a conflict management could then handle how objects are treated depending on where they were found, e.g. if the person on whose desk the object was found works in same project as the searching person and the like.

Developer Workshop and User Evaluation Workshop
This prototype was presented in workshop III with developers and project members working on the training and support of users. They suggested some minor changes concerning the handling and proposed to incorporate the possibility to open a video channel for communication about the search from the search tool dialog as soon as video is available for POLITeam.

The changes were made and the resulting prototype was presented to three users from one of our application partner organizations in workshop IV held at the University of Bonn. Its primary goal was the evaluation of the functionality and user interface of the new search tool. Two of the three users had been interviewed in the initial phase of the redevelopment. By this time the three had used POLITeam intensively for about 10 weeks. We did not just want to give a demonstration of the search tool but provide a chance for hand-on testing. In order to support material for a discussion of the roles of a searcher and a person "being searched on" we prepared five search scenarios. This was done by rebuilding parts of the structure of the desk the users knew from their daily work and providing computers in two separate rooms to represent two users of POLITeam. We planned to have them search the system including other people's desks for a file they needed to proceed with their work and find out what would happen on either of the both computers. Some of the aspects that were meant to be raised by the scenarios were already discussed when we talked about the functionality of the search tool since the three users were experienced and interested enough to recognize what chances and problems might come along with the search tool. They even started a discussion of the different roles of a searcher and a person being searched on by themselves. Thus, it proved to be an advantage that they already had experience with POLITeam so they could well imagine the search tool in their daily work. For example a user imagined his boss working on the computer late at night searching for documents containing certain key words and stressed the importance to be able to create private domains that others could not access with the search tool. After using the search tool for some of the scenario searches we had prepared and some searches initiated by themselves the users made concrete suggestions on how to improve the input dialog in stating that they usually did not know what a certain person had to do with an object, i. e. if she was the owner, creator or had changed the object, only that she had some relation to it. So they suggested that in the search dialog section where the creator, owner and changer of a document could be specified there should not be three entries but just one so that a person could be specified as having to do with an object with the option to say if she was the owner, creator or changer if you knew. Thus, the former need to put in three times the same name for creator, owner and changer and connect them with a logical OR is reduced to just pick one name.

State of work
After prototype 2 has been discussed in workshop IV the changes to it suggested by the group of POLITeam users will be made so that the resulting software will be ready for release with the next POLITeam version. With the search tool introduced then the three main problems that arose with the original search tool (unintended deletion of files, user interface, conflict potential) will be solved or prepared to be solved after a process of participatory and evolutionary software development. Moreover, by new ways of representing objects in a hierarchical tree-like structure and with the chance to order objects by different criteria the refinding of objects on user's own desk is considerably improved.

First important steps for the system's conflict handling are made. The new search tool incorporates some prerequisites of conflict detection in showing where the objects were found before they are picked for the search window. The conflict potentials caused by the activation of POLITeam's group-related functionality require a special module for conflict management for POLITeam which can then be used by the search tool and which will provide ways to detect and solve conflicts e. g. by informing someone that their desk is searched or giving them the chance to veto against it.

DISCUSSION

The course of the development of a search tool for POLITeam has shown that an evolutionary and participatory approach for the development of groupware is promising. The different participatory techniques used brought different insights:

- Feedback from the POLITeam users to the user advocates showed aspects of their cooperative work practice.
- Interviews helped to understand how people search at their workplace and what the requirements for a search tool from the viewpoint of persons searching and "being searched on" might look like.
- Workshops with POLITeam users brought up group-related aspects of system use and increased the users' and developers' understanding of conflicts raised by system use.
- A special workshop to present the search tool prototype to POLITeam users and have them evaluate it in a first step allowed for fine-tuning the search tool to the needs of the application partner and hands-on experience helped to deepen the users' understanding of the conflict potential on a more concrete level. Here, we particularly profited from the fact that the three users were very interested and above-averagely competent in working with POLITeam.

However, our activities would have benefited from a workshop particularly focusing on the potential conflicts of searching on other persons' electronic desks and discussing the implications with a group of POLITeam users at an early stage of the development process. This could have helped the developers to learn about the handling of this issue in a concrete organization and give hints for the implementation while users could have become more aware of the implicit rules of their organization and the technical potential to reveal and support them. Unfortunately, the limited amount of time on the application partners' side and the resources provided for the development of the search tool as only one of many of the POLITeam activities did not allow for such a workshop.

Moreover, the decision to develop a new search tool rather than improving the existing search tool mainly depended on technical considerations. While LinkWorks™ provides some mechanisms to modify or enhance the system's functionality these mechanisms are still not flexible enough since they impose restrictions to the desired implementation.

The STEPS approach (see fig. 1) taken as a basis proved to be helpful as a rough guideline for development. Unlike described in the STEPS model and unlike most of the activities within POLITeam the redevelopment of the search tool was not preceded by the usage of the respective functionality of LinkWorks™ since this was considered to cause too many problems for the application partners. The development activities described above can be located in the production phase of the STEPS model. If we had decided to change the original search tool to fit the new requirements rather than redeveloping it the activities might have been considered to have more of an *adaption* than of a *production*. In that case the development activities could have been located in the application phase of the STEPS model enhanced by the common activity of adaption as suggested by Wulf & Rohde (1995).

CONCLUSION

Two more general aspects of introducing the search tool within the POLITeam framework deserve attention. One of them is the usage of participatory design methods and techniques for the introduction of groupware functionality. Our case study supports the notion that both group workshops and having end users take roles as activator of and someone being affected by a groupware function help to create cooperative awareness. Thus, the concept of perspectivity originally meant to bridge the gap between users and developers is enhanced to let end users get an impression of how other groupware users are affected by their use of functions. At the same time this helps to understand the actual work practice and to make explicit who may cooperate with whom in which way.

The second aspect that our case study contributes to is the introduction of a *generic* groupware product into an organization. Considering the growing need for technology for cooperation and communication inevitably most of the groupware applications installed and used in the future will be generic applications that are adapted to the needs of a special organization. The usability and success of this groupware will to a large degree depend on the quality of this adaption.

Generally the disadvantage of a commercial off-the-shelf product is that it ignores specific social and organizational concerns and users are not known at the time of initial development (Grudin 1991). By providing both organizational means to introduce the groupware and technical mechanisms that allow for different levels of tailoring (cf. e.g. Henderson & Kyng) this disadvantage might be overcome. On the technical side our experiences lead to the conclusion that the approach taken by LinkWorks™ which is based on object-orientation and provides an application programming interface as well as means to change internal methods looks promising. Still, the mechanisms of LinkWorks™ were not flexible enough to fulfill all our needs. Object orientation also plays an important role in more detailed and concrete suggestions made concerning technical means to support tailorability (cf. e.g. Fischer & Girgensohn 1990, Malone et al. 1992, Mørch 1995). The need for flexible solutions also includes the demand to allow for unanticipated use by supporting the notion of the medial character of the groupware and avoiding the implementation of rigid user "representations" (Bentley & Dourish 1993).

Probably most important for the introduction of generic groupware is an adequate organizational treatment. Previous work on the area of introducing generic groupware into an organization has shown the need for explicit organizational embedment in order to use the full range of groupware advantages (Orlikowski 1992) and drawn the attention to the interplay of intended and emergent induced organizational changes by groupware use (Orlikowski 1995) that demand technical flexibility. These organizational changes will be analyzed carefully in the POLITeam project to learn more about the impact of introducing a groupware and have the introduction process benefit from this knowledge.

In many ways introducing a generic groupware resembles the design process for the development of a custom-made groupware. Here, methods of participatory design can be used for participatory tailoring. Research having taken into account the influence of a group structure for participatory design and development can give important hints for methods of participatorily introducing and tailoring generic groupware. For example, Kjær & Madsen (1994) suggest a participatory analysis of flexibility based on a "blueprint mapping" technique to get an overview of the daily work and on an "organizational game" to analyze the need and potential for organizational flexibility. Another closer look at organizational aspects of tailoring that can go beyond the phase of initial implementation is taken in some papers dealing with the sharing of customization files (Mackay 1990; Nardi 1993 Chapter 6; Trigg & Bødker 1994). While these findings are not explicitly related to groupware they involve group activity to customize software used by a group. The papers stress the importance of local experts who know the work practice well enough to provide adequate customization.

Still, more work has to be done on the impact of group particularities on the use of groupware functionality, how roles are represented in groupware and how conflicts can be detected and mediated that are induced or made visible by system use. Here, many questions remain open (cf. Kahler 1995). How can we proceed when introducing one groupware for different organizations? How much tailoring can and must be done? What can participation not only in the process of design but also in the process of introducing a system look like? How can we share responsibilities for customizing groupware for an organization between users and developers? How can we support participation for system introduction and customization by preconfigured systems?

This paper has provided a case study indicating that continuous work with an evolutionary and participatory approach to the development of groupware and its introduction may help to answer these questions.

ACKNOWLEDGEMENTS

I would like to thank Guido Krüdenscheidt who implemented the search tool, our partners from the interviews and workshops for contributing their time and patience, and the three anonymous PDC reviewers for helpful comments on an earlier version of this paper.

REFERENCES

Barreau, D.; Nardi, B. A. (1995): Finding and Reminding. *SIGCHI Bulletin July 1995*, pp. 39-43

Bentley, R.; Dourish, P. (1995): Medium vs. Mechanism: Supporting Collaboration Through Customization. In: *Proceedings of the ECSCW '95*, Kluwer, pp. 133-148

Boehm, B. W. (1976): Software Engineering. *IEEE Transactions on Computers, Vol. C-25, No. 12*, December 1976, pp. 1226-1241

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

Ehn, P. (1993): Scandinavian Design: On Participation and Skill. In: Schuler, D.; Namioka, A. (eds.): *Participatory Design - Principles and Practices*. Lawrence Erlbaum, Hillsdale, NJ, pp. 41-77

Fertig, S.; Freeman, E.; Gelernter, D. (1996): "Finding and Reminding" Reconsidered. *SIGCHI Bulletin January 1996*, pp. 66-69

Fischer, G.; Girgensohn, A. (1990): End-User Modifiability in Design Environments. In: *Proceedings of CHI '90*, pp. 183-191

Floyd, C.; Reisin, F.-M.; Schmidt, G. (1989a): STEPS to Software Development with Users. In: Ghezzi, C.; McDermid, J. A. (eds.): *ESEC '89 - 2nd European Software Engineering Conference*. pp. 48-64

Floyd, C.; Mehl, W.-M.; Reisin, F.-M.; Schmidt, G.; Wolf, G. (1989b): Out of Scandinavia: Alternative Approaches to Software Design and System Development. *Human-Computer Interaction, 1989*, Vol. 4, pp. 253-350

Floyd, C. (1993): STEPS - A Methodical Approach to PD. *Communications of the ACM, Vol. 36, No. 4 (June 1993)*, p. 83

Greenbaum, J.; Kyng, M. (eds.) (1991): *Design at Work. Cooperative Design of Computer Systems*. Hillsdale, NJ

Grønbæk, K.; Kyng, M., Mogensen, P. (1995): Cooperative Experimental System Development - cooperative techniques beyond initial design and analysis. In: *Computers in Context: Joining Forces in Design*. Aarhus pp. 20-29

Grudin, J. (1991): Interactive Systems: Bridging the Gaps Between Developers and Users. *IEEE Computer April 1991*, pp. 59-69

Henderson, A.; Kyng, M. (1991): There's No Place Like Home: Continuing Design in Use. In: Greenbaum, J.; Kyng, M. (eds.): *Design at Work. Cooperative Design of Computer Systems*. Hillsdale, NJ. pp. 219-240

Henderson-Sellers, B. and Edwards, J. M. (1990): The object-oriented System Life Cycle. *Communications of the ACM, Vol. 33, 9*. pp. 143-159

Hesse, W.; Weltz, F. (1994): Projektmanagement für evolutionäre Software-Entwicklung. *Information Management 3/1994*, pp. 20-32

Kahler, H. (1995): From Taylorism to Tailorability - Supporting organizations with tailorable software and object orientation. In: Anzai, Y.; Ogawa, K.; Mori, H. (eds.): *Symbiosis of Human and Artifact, Proceedings of HCI '95, Vol. 20B*, Elsevier, Amsterdam, pp. 995-1000

Kjær, A; Madsen, K. H. (1994): Participatory Analysis of Flexibility: Some Experiences. In: *Proceedings of the PDC '94*, pp. 21-31

Klöckner, K. et al. (1995): POLITeam -Bridging the Gap between Bonn and Berlin for and with the Users. In *Proceedings of the ECSCW '95*, Kluwer, pp. 17-32

Krüdenscheidt, G. (1996): *Partizipative Entwicklung eines Suchtools für Groupware*. Diploma Thesis, Bonn (in German; available from the author)

Lehman, M. M.; Belady, L. A. (1985): *Program Evolution: Processes of Software Change*. London

Mackay, W. E. (1990): Patterns of Sharing Customizable Software. In: *Proceedings of the CSCW '90*, pp. 209-221

Malone, T. W.; Lai, K.-Y.; Fry, C (1992): Experiments with Oval: A Radically Tailorable Tool for Cooperative Work. In: *Proceedings of the CSCW '92*, pp. 289-297

Mambrey, P.; Mark, G.; Pankoke-Babatz, U. (1996): Integrating User Advocacy into Participatory Design: the Designer's Perspective. In: *Proceedings of the PDC '96*, in this volume

Mørch, A. (1995): Three Levels of End-user Tailoring. In: *Computers in Context: Joining Forces in Design.* Aarhus pp. 157-166

Nardi, B. M. (1993): *A Small Matter of Programming.* Cambridge, Massachusetts, MIT Press

Orlikowski, W. (1992): LEARNING FROM NOTES: Organizational Issues in Groupware Implementation. In: *Proceedings of the CSCW '92*, pp. 362-369

Orlikowski, W. (1995): EVOLVING WITH NOTES: Organizational Change Around Groupware Technology. *CCS Working Paper 186, MIT, Sloan School of Management*, Cambridge, MA; URL: http://ccs.mit.edu/ccswp186.html

Trigg, R.; Bødker, S. (1994): From Implementation to Design: Tailoring and the Emergence of Systematization in CSCW. In: *Proceedings of the CSCW '94*, pp. 45-54

Wulf, V.; Hartmann, A. (1994): The Ambivalence of Network's Visibility in an Organizational Context. In: Clement, A.; Kolm, P.; Wagner, I. (eds), *NetWorking: Connecting Workers In and Between Organizations*, North Holland, Amsterdam, pp. 143-152

Wulf, V. (1995): Mechanisms for Conflict Management in Groupware. In: Anzai, Y.; Ogawa, K.; Mori, H. (eds): *Symbiosis of Human and Artifact, Proceedings of HCI '95*, Vol. 20A, Elsevier, Amsterdam, pp. 379-385

Wulf, V.; Rohde, M. (1995): Towards an Integrated Organization and Technology Development. In: *Proceedings of the Symposium on Designing Interactive Systems*, 23.-25.8.1995, Ann Arbor (Michigan), ACM-Press, New York, pp. 55-64