in: Proceedings of GROUP 2001, ACM-Press, New York 2001, pp. 107 - 116

# Exploration Environments – Concept and Empirical Evaluation

Volker Wulf
GMD-FIT: National Research Center for Information Technology
Schloß Birlinghoven, 53754 Sankt Augustin, Germany
volker.wulf@gmd.de

Björn Golombek
IISI: International Institute of Socio-Informatics
Schützenstr. 3, 53111 Bonn, Germany
bg@intra.de

## ABSTRACT

Exploration environments support users to learn groupware in a self-directed way. They allow users to perceive the way a groupware function works by simulating the impacts of the function's execution on other users' interfaces. This paper motivates and describes the concept of exploration environments and presents an exemplary implementation. Moreover, it reports about the results of an experimental study, which evaluated the effects of this concept on users' learning. The results of this study are discussed and compared to findings from a field study which evaluated the concept, as well.

## Keywords

Learning, Exploration, Groupware, Tailoring, Evaluation

## 1. INTRODUCTION

Suitability for learning is a key principle for the design of the user interface of any interactive system (cf. ISO 9241 part 10). Looking at the learning of a computer application, there are two different ways how individuals can increase their knowledge about its functionality. On the one hand the learning process can be structured externally. This is typically the case with computer courses, seminars, and manuals. On the other hand users can learn an application by self-directed learning activities (cf. Dutke 1994, pp. 146). The results of different empirical studies show the importance and efficiency of self-directed learning activities (cf. Carroll and Mack 1983 and 1984, Kamouri et al. 1986, Greif and Jankowski 1987, Frese et al. 1988).

Self-directed learning can be supported by the design of the user interface. Focussing on single user applications, Paul (1994, pp. 156) gives a survey on features which promote self-directed learning. He distinguishes two main activities: investigating the functionality of an application and experimenting with certain of its functions. During the investigation the user tries to get an overview of the given functionality of an application. Survey functions, histories of interaction, and scenario machines support this phase of the learning process. While experimenting with a certain function, the user tries out which state transition follows the execution of this function. Hence undo-functions allow to reset the execution of a function while freezing points allow to define certain system states in advance to which users can come back after having tried out other functions. Experimental data are specifically created to support users in experimenting with certain functions. A neutral mode replaces the execution of a function by a textual description of the effects of its execution (cf. Yang 1990; Howes and Paynes 1990; Paul 1994).

In this paper we focus on technical features which support end users in experimenting with functions in groupware. Groupware is used as a general term for multi-user applications, which support communication, coordination, and cooperation among their users (cf. Schmidt 1991; Wulf 1997). This definition implies that the execution of groupware functions may have effects on other users' interfaces. When a user experiments with a groupware function he may not be able to fully perceive the effects of its execution. This fact poses new research challenges.

Let us take the example of a mail tool: A user who wants to learn about the forwarding function can not fully perceive the effects of the function's execution on his own interface. For instance, the learner may want to know: (1) whether he has specified the forwarding function correctly (e.g. the address of the recipient), (2) how the forwarded mail is indicated on the side of the recipient (e.g. received in which mail box? presented by means of which attributes?), (3) whether attachments are forwarded and how they are displayed. To get this information which are essential for learning, he has to observe the recipient's user interface. However, the recipient may be far away or he may feel disturbed by other user's learning activities.

So it is a completely different learning situation compared to functions in single user applications. If these applications are designed according to the principles of direct manipulation (cf. Shneiderman 1983), instant feedback about the effects of a function's execution will be given to the learner on his own interface. Therefore, groupware requires new concepts to support self-directed learning.

The problem of learning gets even more difficult when the functions of the groupware are tailorable to a wide extend (cf. Oberquelle 1994; Bentley and Dourish 1995, Kahler et al. 2000).

In this case, users are equipped with different tailoring functions, which extend or modify the functionality of an application by creating or modifying persistent artifacts. The execution of tailorable functions depends on the state of the corresponding tailored artifacts. So when experimenting with a tailorable function, the learner has to understand the actual state of the function.

With regard to learning processes of tailorable groupware, the distinction between normal and triggered functions is of especial importance. Normal functions are typically represented at the surface interface of an application, activated by an input of the user at one of their access points, and executed in the way prespecified by the tailoring function or specified during its activation. Looking at graphical user interfaces according to the principle of direct manipulation, a normal function can be typically activated at the location where it is displayed on the screen (e.g. in case of menu item, button, or icon). Nevertheless, there might exist other access points which are not represented on the screen (e.g. short cuts). Triggered functions are not represented at the surface interface. They are activated by the occurrence of pres pecified events which are either created internally or result from the interaction with external applications. A triggered function is executed in the way prespecified by the corresponding tailoring function (cf. Oppermann and Simm 1994, Wulf 1999a). An example of a normal function is “sending mail“ while the “mail filter“ is a function triggered by an event, i. e. the incoming mail.

Due to the multi-user character and the high level of required tailorability, the design of groupware poses new challenges, when supporting self-directed learning of normal and triggered functions and their corresponding tailoring functions. To ease experimenting with groupware functions we will present the concept of exploration environments.

This paper motivates and describes the concept of exploration environments and shows an exemplary implementation. Moreover, it reports about the results of an experimental study which evaluated this concept. The paper is organized in the following way: In the next section we will summarize the results of the empirical prestudies and develop the concept of exploration environments. To find out whether these concepts support learning of tailorable groupware, we have conducted a test-statistical evaluation. The results of this investigation will be presented and compared to those of qualitative evaluations within an organization.

## 2. Exploration Environments – Motivation and Concept

Empirical prestudies in the context of the POLITeam project (cf. Pipek and Wulf 1999) have shown that there exist specific problems when users try to learn groupware functions. Most of these problems did not only appear within the introduction phase of the groupware. They remained even after a long-term use of the system. We noticed the following learning problems:

- The effects of executing a function are hard to understand if only seen from the perspective of the learner’s interface. For instance, the difference between sending a document and sharing a document was hard to understand when seen only from the interface of the user who wanted to distribute the document (cf. Wulf 1999b).
- In case the function to be learned is tailorable, users have additional problems to perceive the actual state of a function and to understand the differences between alternative ways to execute this function. For instance, users had problems to understand the differences among the alternatives of a highly tailorable search tool (cf. Wulf 1999c).
- Additional learning problems arise, if a function’s execution is influenced by triggered functions. This problem gets worse if the triggered functions are tailored from other users’ interfaces. For instance, users had considerable problems exploring a search tool in groupware because they did not know whether lacking search results were due to a faulty specification of their inquiry or missing search permissions granted to them by other users (cf. Wulf 1999c).
- The execution of triggered groupware functions often requires an event generated from another user’s interface. So these functions can only be explored if the user, who wants to learn, has access to other users’ interfaces. For instance, a complex access profile granted to a document was difficult to understand. It was necessary to try it out from the interface of those users to whom the specific access rights had been granted (cf. Wulf 1999b).
- Collaborative learning of groupware occurs especially during the introduction phase when actors are interested in exploring the application. If the users’ skills differentiate during the use of an application, such learning activities may be regarded as disturbances by other users. For instance, a newly employed secretary had considerable problems finding partners to experiment with the groupware (cf. Wulf 1999b).

To support users in overcoming the problems mentioned above, we developed the concept of exploration environments. An exploration environment allows simulating the execution of groupware functions by means of a specific system mode. In this mode, the learner’s own user interface, as well as the behavior of other users’ interfaces, are simulated on the learner’s output device. If a function is executed in an exploration environment, the effects of its execution on the simulated user interfaces will be similar to the effects the “real” function’s execution has on the “real” user interfaces. By executing a function in the exploration environment and switching between the simulation of his own and the other users’ interfaces, a user can learn how a certain function works.

The other users’ interfaces may be simulated graphically in the exploration environment. In this case the simulated interfaces must be designed in a way closely related to their appearance in the real system. So, the learner gets a “look and feel“ of the effects of a function’s execution by interacting with the other users’ simulated interfaces. He may also be able to execute a function on the other users’ simulated interfaces. This may be necessary if he wants to learn about a function which is triggered by other users’ activities (e.g. mail filter).

## 3. Exploration Environments – A Design Study

In the following we will present the example of exploration environments which support users in understanding the way filters of a tailorable awareness service work. The tailorable awareness service is implemented based on a groupware tool which supports cooperative tailoring activities. This groupware tool extends the MS-Word application. Each user of the tool has a private workspace to store tailored artifacts and a mailbox to receive tailored artifacts sent by other users. Moreover, a shared workspace is provided to which users can publish tailored artifacts from their private workspaces, and from which they can copy tailored artifacts to their private workspaces. This tool has been developed in VBA (Visual Basic for Applications) and integrated in the MS-Word menu bar (cf. Kahler et al. 1999).

To inform users about events concerning shared tailored artifacts, an awareness service has been implemented in the tool. Eight different types of events are automatically recorded, distributed among the users and displayed at their interfaces. To avoid users being overloaded when receiving useless messages and to protect their privacy, the awareness service is tailorable by the realization of two filters.

The recipient's filter allows selecting the relevant events. Therefore, the users can specify which type of event (out of the given list of event-types) generated by whom (out of the list of users and groups of users) should be displayed with a certain intensity (pop up window, icon presentation in the status bar, or not at all). To ease the tailoring process, we provided two tailoring functions of different levels of complexity. On the highest level, users could select an interest profile for each event-type. This profile is selected out of a list of given profiles. It indicates in plain text whose events should be displayed in which intensity. On the second level of complexity the users can generate new interest profiles by combining certain creators with display intensities.

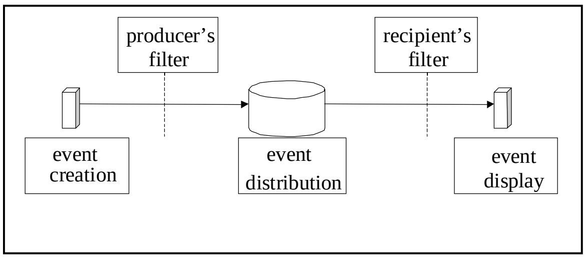


**Figure 1: Architecture of the tailorable awareness service**

The producer's filter allows the user to decide which of their events should be published to whom. Here again we provide two different layers of tailoring complexity. On the first level the users can decide which event should be distributed to which user or user group. On the second level of tailorability the users can generate new groups of users. The recipients can display only those events which had been published by the producer. Compared to existing implementations of similar awareness services (e.g. Fuchs 1997), the privacy filter is realized additionally. Figure 1 gives a survey on the architecture of the tailorable awareness service.

Exploration environments have been implemented concerning the whole tool which extended the MS Word application. In this paper, we will focus on exploration environments concerning the tailorable awareness service.

The recipient's and the producer's filters are both triggered functions which are not represented at the user interface. Therefore, we had to decide how users could find the corresponding exploration environment. We included the access point to the triggered functions' exploration environments into the windows of their first level tailoring function.

The display of an incoming event is designed in a way that the tailoring function of the recipient's filter can be activated directly (cf. Wulf 1999a). We equipped the window of this tailoring function with a button to enter the exploration environment of this filter. So, whenever the users have tailored the filter, they can test the outcome of their activities.

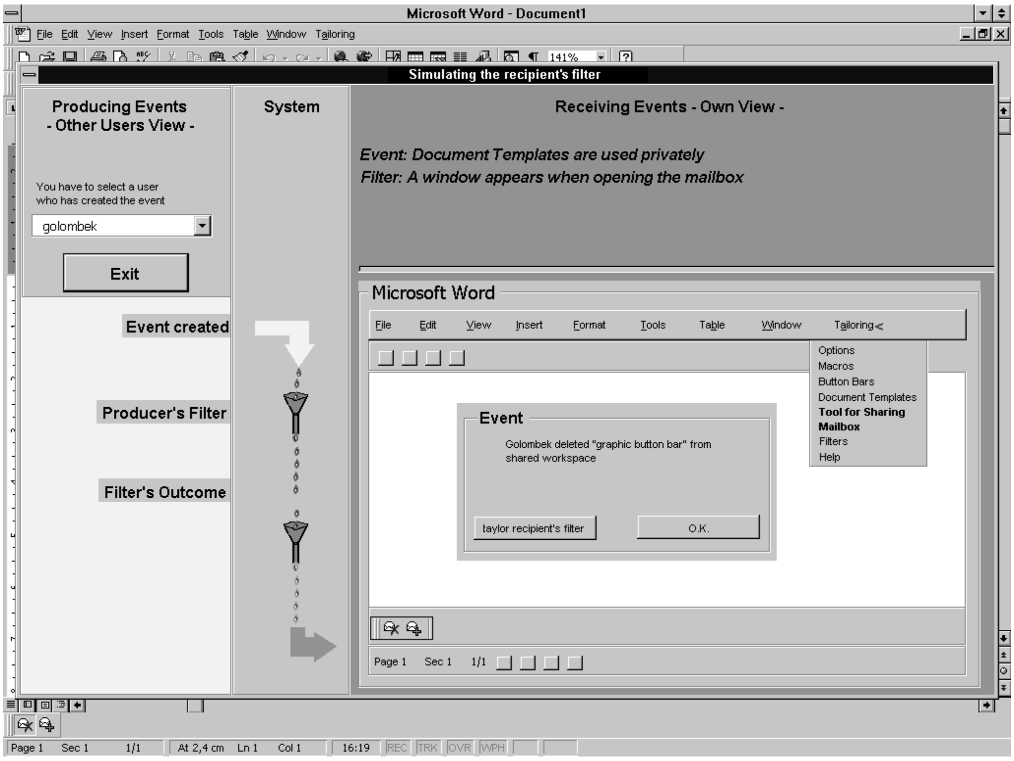
After the exploration environment has been activated, a window appears, which asks the user to specify an event. Afterwards a new window appears which is subdivided into two parts (cf. figure 2). These parts differ in background color and title. The left part represents the simulated workspace of another user. The right part represents the learner's workspace. Moreover in the window there are some gray fields, which contain functions to describe and control the exploration environment (e.g. exit the exploration environment, return to the event specification). By means of a blinking arrow next to the corresponding pull down menu, the learner is asked to attribute the left side of the window to one of the other users.

After an event has been created on the side of the other user, the state transitions caused by the execution of the two filters are visualized. For the visualization we use the metaphor of water dropping through two funnels. A drop of water symbolizes an event, the gravity indicates the awareness service and the two funnels symbolize the filters. Each funnel can be either open or closed. Text marks describe the different stages of the event handling. The drop will pass the first funnel if the producer's filter of the selected user allows publishing this event. It will pass the second funnel if the recipient's filter of the learner allows displaying the event. If the event is eliminated at one of the filters, this fact will be symbolized at the corresponding funnel and described textually.

In case the event has passed both filters, a simulated MS-Word environment can be opened on the right side of the window (learner's side). This environment simulates the learner's functionality. Most of the menu bars are not available except for the one which allows opening the tool to share tailored artifacts and the mailbox to receive them. Hence the learner can check out the different situations in which he might be notified about an incoming event. In the simulated MS-Word environment the events are now displayed with the prespecified intensity (cf. figure 2).

Having explored how a certain activity of another user is displayed on his screen, the user can generate new events created by other users or leave the exploration environment. He returns to the window where the recipient's filter is tailored.

If a user wants to add or edit an interest profile with regard to a certain event, he can activate the corresponding tailoring function directly. To examine newly created interest profiles, another exploration environment can be activated in this window. It is implemented similarly to the one of the first level tailoring function. Nevertheless the learner does not have to specify the event created by the other user. It is already known to the system because the interest profiles are created for one specific event type. So, the learner can start directly by selecting the other user whose event production is simulated in the exploration environment. After leaving the exploration environment the learner returns to the window where the interest profiles can be created.


Microsoft Word - Document1
File Edit View Insert Format Tools Table Window Tailoring
141%
Simulating the recipient's filter
Producing Events
- Other Users View -
System
Receiving Events - Own View -
golombek
Exit
Event created
Producer's Filter
Filter's Outcome
Microsoft Word
File Edit View Insert Format Tools Table Window Tailoring
Options
Macros
Button Bars
Document Templates
Tool for Sharing
Mailbox
Filters
Help
Event
taylor recipient's filter
O.K.
Page 1 Sec 1 1/1
Page 1 Sec 1 1/1 At 2,4 cm Ln 1 Col 1 16:19 REC TRK OVR WPH

*Event: Document Templates are used privately*
*Filter: A window appears when opening the mailbox*

You have to select a user
who has created the event

Golombek deleted "graphic button bar" from
shared workspace

**Figure 2: Environment to explore the recipient's filter**

Like the recipient's filter the producer's filter is a triggered function of which execution has been made visible to the users. In this state the corresponding tailoring function can be activated directly. Like in the case of the recipient's filter, we equipped the window of the first level tailoring function with a button to enter the exploration environment for this filter.

After the exploration environment has been activated, a window appears which is subdivided into two parts. The left part represents the learner's tool for sharing tailored artifacts. The right part represents the workspace of another user. Like in the case of the recipient's filter, there are some gray fields in the window, which describe the exploration environment and contain functions for its control.

By means of blinking arrows next to the corresponding pull down menu, the learner is invited to select an event and a recipient. After the user's input, the button of the simulated sharing tool of which activation creates the selected event is indicated. By pressing this button the user creates the prespecified event. The metaphor of water dropping through two funnels is applied here again to visualize the two filters of the awareness service.

If the event has passed the two filters, the learner can move into the other user's part of the window and activate another window where the MS-Word application is simulated. In this window the learner can try out when and how the event is displayed on the side of the recipient. Afterwards the learner can choose to explore the effects that the awareness service has on workspaces of other recipients. Alternatively he can push a button to leave the exploration environment.

## 4. Discussion: Design Study

When simulating a groupware function one has to realize the function's behavior and its different user interfaces in the exploration environment. Moreover, experimental data have to exist in the exploration environment. The function in the exploration environment should carry out the same state transitions on the experimental data as the original function would do on the original data.

Experimental data may be either copied from the original data or they have to be created additionally. In the later case further functions may have to be implemented in the exploration environment, which allow to generate experimental data. In the case of the tailorable awareness service, the configuration of the filters of the other users are copied if the owners of the filter authorize this activity by setting a corresponding flag. If the owner of the filter does not allow copying, the configuration of the filters are created at random. This fact is indicated to the user in plain text.

When designing an exploration environment for a specific function, one has to decide which other functions are relevant to be included. This decision has to be made concerning other ordinary functions and the different tailoring functions. Concerning other ordinary functions, our design has been guided by the idea that a tailoring environment should be kept as small and simple as possible. As to a triggered function only those other functions were included in the environment which can generate events leading to the function's execution (e.g. a function to create an event on the side of an other user when exploring the recipient's filter). With regard to a normal function, only those other functions were included, which influence the function's execution.

Tailoring functions can be included in, or excluded from the exploration environment of the tailorable functions. If they are included, the tailoring environments get more complex. In case they are excluded additional exploration environments for the tailoring functions have to be created. In the design study presented here we opted for the second alternative.

The different user interfaces of the exploration environment should be implemented related as closely as possible to the interfaces of the original application. However, we used different background colors to support users in distinguishing between the original application and the exploration environment. The fact that many functions of the original application are not part of the exploration environment should be instantly visible to the user. In our design study we used gray letters and icons to indicate non-usable access points.

In order to ease the learner's orientation, it is necessary to clarify the differences between simulated interfaces of the other users and the one of the learner. This was archived by different colors of windows and descriptive texts.

Spatial limitations of the output devices (e.g. screens) make another difference between the user interfaces in the exploration environment and the ones in the real application. They required simplifying the original windows and sequentializing the output of the exploration environment. Moreover, additional functions of the exploration environment have to be presented at the user interface (e.g. the button to exit the environment). It turned out to be helpful to visualize hidden state transitions following the execution of triggered functions (e.g. funnels visualizing the existence of the filters). Especially if the exploration environment is more complex in structure, additional features may be helpful to guide the user's exploration (e.g. the blinking arrows which guide the sequential activities).

## 5. Empirical Evaluation

Exploration environments are supposed to encourage self-directed learning of tailorable groupware. The concept has been motivated by users problems which came up in empirical prestudies. Now, we will evaluate whether these concepts encourage learning by overcoming the mentioned problems. Therefore, we have carried out an experiment and a field study. The experiment evaluated the exploration environments of the tailorable awareness service described in section 3.

Moreover, we have evaluated exploration environments designed for a search tool in groupware. These features have been developed in a cooperative way with users from a body of the German political administration (cf. Wulf 1999c). The final version of these exploration environments has been presented in a workshop.

In the following we will first describe the results of the experiment and then compare the findings of the experiment to the ones of the field study, which have already been published (cf. Wulf 2001).

The concept of exploration environments is supposed to support users in getting a better and faster understanding of the effects of a groupware function's execution. As the exploration environments support the comprehension of tailorable groupware functions, we let users predict the behavior of these functions. Moreover, we were interested in the effects of an improved understanding on the users' task performance. We assumed that an improved understanding would have a positive effect on the outcome and the time efficiency of computer supported tasks. In case of the awareness service, the exploration environments supported users to tailor the different filters. So, we measured how well and how fast users were able to tailor the awareness service.

*Hypothesis 1: Exploration environments lead to a better understanding on how the execution of a groupware function works.*

*Hypothesis 2: Exploration environments lead to a better tailoring performance.*

*Hypothesis 3: Exploration environments lead to a faster tailoring performance.*

In order to test these hypotheses, we have developed two different versions of the tailorable awareness service presented in section 3. Exploration environments had been implemented in the first version of the tool. The second version of the tools did not contain these environments. By means of an experiment we compared the performance of the users working with the different versions of the tool (experimental conditions). This experiment was part of a larger study in which other concepts to encourage self-directed learning had been evaluated, as well.

### 5.1 Subjects

There were 22 subjects, 11 for each experimental condition. Their age ranged from 20 to 35 years. About 80% of the subjects were students with a variety of specialties. Moreover, several office workers took part in the study.

We checked the subjects' computer experience before the experiment by means of a questionnaire. Based on the result of this investigation, we selected a rather homogeneous group of subjects. All of them had used word processors before, while none of them had experiences being a programmer. We also documented further aspects of their computer experience (e.g. tailoring experiences, use of e-mail). The subjects were assigned to the two experimental conditions at random. Those subjects who had been equipped with the system containing the exploration environments will be called the EA group, the other group is called the control group. At the end of the experiments all subjects took part in a lottery of which the prize was two hundred German Marks.

## 5.2 Experimental Procedure

The experiment took place in the research lab of the University of Bonn. The lab had been equipped with two networked computers facing each other. On both of the computers the groupware tool was running. The subjects were asked to use the first computer while the second computer was used by one of the experimenters during the experiment. The subjects had been informed that the same tool was running both of the computers. During the whole experiment the subjects could move to this computer, watch the screen, or ask the experimenter to give them access. So the experimental setting offered access to a second interface to explore the awareness service. We assumed that such a setting was realistic because in office settings users often had access to a second computer on which the same groupware application ran.

The tool did not contain a help function. The subjects could ask the experimenter whenever they had questions concerning the tool. However, the experimenter did not answer questions related to the given tasks.

The experimental setting offered another feature to support self-directed learning. The producer's filter could be tailored in a way that the creator received his own events. The subjects of both experimental conditions had the chance to explore the awareness service by means of this feature.

The test consisted of two phases, which were almost identically in both of the experimental conditions. In the first phase the subjects were asked to explore the functionality of the tool while in the second phase they had to work on given tasks. The experimenter started the first phase of the experiment by giving an overview of the given version's functions. This overview was identical for both experimental conditions, except that the EA group got a brief presentation of the exploration environments, as well.

To direct the learning activities towards those functions, which were relevant for the second phase of the test, some tentative tasks had been given to the subjects. The subjects of both groups got the same tasks. The task performance was not checked during the first phase of the experiment. The subjects could decide when to finish with the first phase of the experiment. The experimenter stopped this learning phase after a maximum time span of half an hour. Except for two subjects all of the others made full use of the given time for learning. During the first phase, we documented the use of the exploration mode and the access to the different tailoring functions.

In the second phase of the experiment the subjects had to solve five tasks. The performance of these tasks was supposed to indicate the effectiveness of learning. The first two tasks asked the subjects to modify the filter settings by means of the given tailoring functions. The other three tasks asked the subjects to predict the behavior of the tailorable awareness service. We started the second phase of the experiment by reconfiguring the awareness service in a similar way for all subjects. There was no time limitation at all for the second phase of the experiment. The subjects could work on the tasks as long as they wanted.

The first task invited the users to modify the intensity for the display of a certain type of event. One of the experimenters created such an event. Due to the preconfiguration of the awareness service, it was displayed on the subjects' screen at the intensity level of a „pop-up window". The subjects were asked to reduce the intensity of display to „icon presentation in the status bar". The second task asked the subjects to modify the setting of the recipient's filter, again. However, this time it was necessary to create a new interest profile. So the subjects had to activate two different tailoring functions. First they had to create a new interest profile and then select this profile.

> Please set up your recipient's filter in a way that events of the type "copy a document template from the shared to the private workspace" are indicated as an icon in the status bar.

**Figure 3: Description of task 2**

We coded the outcome of the tailoring activities as follows: A one indicated that the tailoring task had been carried out successfully, a zero indicated that the given task had not been carried out successfully.

Three more tasks checked the subjects' overall understanding of the functionality. The subjects were asked to predict the behavior of the tailorable awareness service. While tasks three and four invited the users to predict the behavior of the recipient's filter, task five dealt with the producer's filter. Task three asked the subjects to predict whether and how they would be notified about a document template mailed to them. Task four invited the subjects to predict how recipients got aware of the fact that a tailored button bar had been sent to them. Figure 4 presents the description of task four. Task five asks the subjects to predict whether and how other users would become aware of the fact that they had loaded a button bar from their private folder. A predefined state of their producer's filter and of the other users' recipient's filter was given in the task description.

*You send the button bar "project x" to the user "golombek". How can he perceive this fact?*

*O not at all because ......*

*O he finds it in his private work space because .....*

O he finds it in his mailbox because ......

O by means of the awareness service, in case .......

**Figure 4: Description of task 4**

For each of the three tasks one or more predictions had to be selected out of a list of given alternatives. In some cases, the subjects had to give a brief explanation why they had chosen a certain alternative. Depending on the difficulty to predict the behavior of the awareness service, each correctly selected alternative was counted as either one or two points. Each correct explanation was worth another one or two points. An incorrectly selected alternative or a false explanation reduced the amount of points accordingly. Following this scheme the subjects could gain a maximum of three points in the third task, a maximum of four points in the forth task and a maximum of seven points in the fifth task (cf. table 1). The results of each task were scaled down to the [0, 1]-interval by dividing the points gained through the maximum amount of points to be gained per task.

| | |
|---|---|
| Task 1 | Tailoring: $x \in \{0, 1\}$ |
| Task 2 | Tailoring: $x \in \{0, 1\}$ |
| Task 3 | Understanding: $x = y / 3$<br>$y \in \{0, 1, 2, 3\}$ |
| Task 4 | Understanding: $x = y / 4$<br>$y \in \{0, 1, 2, 3, 4,\}$ |
| Task 5 | Understanding: $x = y / 7$<br>$y \in \{0, 1, 2, 3, 4, 5, 6, 7\}$ |

**Table 1: Points to be gained per task**

While working on these three tasks, the subjects were still sitting at their computer and could use the system to find out appropriate solutions.

Looking at our hypothesis, the first two tasks were related to hypothesis 2, while the third to fifth task were designed to test hypothesis 1. As we assumed that each task had the same importance for testing the hypothesis, we weighted them equally. So we added up the scaled outcomes of tasks 1 and 2 (to test hypothesis 2) and of tasks 3, 4 and 5 (to test hypothesis 1) (cf. table 2).

## 6. Empirical Results

In this section we will present the results of the empirical investigation. Table 2 gives a survey on the subjects' tailoring performance concerning the different tasks and their aggregated performance. We used this data to test hypotheses 2 (exploration environments lead to a better tailoring performance). The aggregated performance is the sum of the performances of the two tasks. Overall, it turned out that the EA group obviously has carried out the tailoring tasks better. This effect is about equally strong for both of the tasks.

Table 2 also presents the subjects' performance on tasks 3, 4 and 5 and their aggregated performance. We used these data to test hypothesis 1 (exploration environments lead to a better understanding of the functions). For each of the tasks the results are scaled down to the [0, 1] interval (cf. section 4). The aggregated performance is the sum of the performances of the three tasks. Overall, it turned out that the EA group obviously has built up the better understanding of the tailorable awareness service. This effect is strongest for tasks 4 and 5.

Table 3 presents the time needed for the five tasks measured in minutes. We used the data concerning the tailoring performance (first row) to test hypotheses 3 (exploration environments lead to a faster tailoring performance). With regard to the tailoring tasks, the EA group was faster in carrying them out. The tasks which tested the subjects' understanding of the tailorable awareness service were also carried out more time efficiently by the EA group. This group had also reached the better results in performing the corresponding tasks (cf. Table 2).

| | Corresponding Tasks | Control Group | | EA Group | |
|---|---|---|---|---|---|
| | | m | sd | m | sd |
| Tailoring Perfor-mance | Task 1 | 0.55 | 0.52 | 0.82 | 0.40 |
| | Task 2 | 0.09 | 0.30 | 0.36 | 0.50 |
| | Tasks 1 + 2 | 0.64 | 0.67 | 1.17 | 0.75 |
| | | | | | |
| Under-standing of the Functions | Task 3 | 0.82 | 0.23 | 0.88 | 0.22 |
| | Task 4 | 0.55 | 0.31 | 0.75 | 0.30 |
| | Task 5 | 0.70 | 0.34 | 0.94 | 0.22 |
| | Tasks 3 + 4 + 5 | 2.07 | 0.69 | 2.57 | 0.57 |

**Table 2: Mean values and standard deviations of task performance**

To test the hypotheses we carried out single tail t-tests between the two groups (cf. table 4). Concerning the t-tests, we asked for a significance level of $p < 0.05$ to accept a hypothesis (cf. Bortz 1993).

| Time needed (min) | Control Group | | EA Group | |
|---|---|---|---|---|
| | m | sd | m | sd |
| Task 1 + 2 | 9.55 | 4.16 | 6.64 | 3.41 |
| Task 3 + 4 + 5 | 20.00 | 7.49 | 14.55 | 4.78 |
| Tasks 1+2+3+4+5 | 29.55 | 10.31 | 21.18 | 7.10 |

**Table 3: Mean values and statistical deviations of time needed to carry out the different tasks (in minutes)**

To test hypothesis 1 (exploration environments lead to a better understanding of the functions), we compared the EA group and the control group with regard to the aggregated results of the tasks 3, 4 and 5. The results of the t-test show significance. The EA group gained a better understanding of the tailorable awareness service than the control group. The results can be interpreted in two different ways: Either the exploration environments had allowed for a better learning of the functionality during the first phase of the experiment, or the use of the exploration environments during the second phase of the experiment better supported the users. Our qualitative observations during the test and the higher time efficiency in solving tasks 3, 4 and 5 rather hint to the second interpretation (cf. table 3). As indicated in table 3, the EA group carried out the tasks 3, 4 and 5 significantly faster than the control group (t (20) = -2.03, p = 0.029). The findings support our hypothesis that exploration environments encourage self-directed learning and lead to a better understanding of the way groupware functions work.

| hypotheses | group | m | sd | group | m | sd | df | t-value | p < 0,1 |
|---|---|---|---|---|---|---|---|---|---|
| H 1 | EA Gr. | 2.57 | 0.57 | C Gr. | 2.07 | 0.69 | 20 | 1.85 | 0.039$^+$ |
| H 2 | EA Gr. | 1.18 | 0.75 | C Gr. | 0.64 | 0.67 | 20 | 1.79 | 0.044$^+$ |
| H 3 | EA Gr. | 6.64 | 3.41 | C Gr. | 9.55 | 4.16 | 20 | -1.79 | 0.044$^+$ |

$^+$ significant ($p < 0.05$), $^{++}$ highly significant ($p < 0.01$)

**Table 4: Results when testing the given hypotheses**

To test hypothesis 2 (exploration environments lead to a better tailoring performance) we compared the tailoring performance of the EA group with the one of the control group concerning the aggregated tasks 1 and 2. The results of the t-test show significance, as well (cf. table 4). This finding supports our hypothesis that a better understanding of the function and its alternatives also leads to a better tailoring performance concerning the respective function.

To test hypothesis 3 (exploration environments lead to a faster tailoring performance), we compared the time spent by the EA group and the control group in carrying out the tailoring tasks. The results show significance in supporting the hypothesis (cf. table 4). This result is of special interest. While it is less surprising that exploration environments lead to a better tailoring performance, they obviously also fasten the tailoring process. So the subjects of the EA group were not only better but also faster in solving the tailoring tasks. Explorative execution such seems to have a positive effect on the efficiency in handling tasks supported by the function to be learned.

## 7. Discussion: Empirical Evaluation

Although the results of the single tail t-tests support our hypotheses, there are some limitations to these results. We carried out the experiment with a rather small number of subjects. Due to the quite high variance in the subjects' individual performance, this fact limited the statistical power of the results. While all results pointed into the same direction, some individual task performances do not show statistical significance (e.g. task 3). Moreover, we selected the subjects very specifically. All of them had already some computer experience. It is unclear whether the given concepts would have encouraged self-directed learning of computer novices or programmers in a similar way.

Another limitation to the results arises from the groupware on which the test was based and the specific design of the exploration environments. We implemented exploration environments concerning a tailorable awareness service of a specific groupware tool. Further investigations are needed to find out whether the positive effects on learning can be found with regard to other groupware functions and applications, as well. Moreover, the concept of exploration environments is formulated in a rather general sense. In this study we have evaluated just one possible way to implement it. However, one can think of other implementations which differ from the ones evaluated. For instance, the event distribution could have been visualized by another metaphor. This means that further research is needed on different variations of this concept.

As with all laboratory studies, one has to be concerned about the external validity of the results. Learning processes in real organizations happen under much more complex circumstances than those given in this study. We will have to find out how exploration environments will be used in the course of the users' daily work.

Finally, there is a need for long-term studies on the use of exploration environments. In the experiment we have just examined the use of exploration environments during the initial learning phase of a groupware. However it is important to find out whether these features influence the task performance in the long run.

In order to compensate for some of the limitations of this study, we have realized exploration environments for two other groupware applications, as well. Their evaluation in real organizations provided us with additional qualitative data. We have carried out a cooperative design project for a tailorable search tool in groupware with users from a body of the public administration. This search tool had been equipped with exploration environments, as well (cf. Wulf 1999c). The results gained within the cooperative design process go mainly along with the findings of the experimental study presented here (cf. Wulf 2000). Especially with regard to complex search tool alternatives, users found it helpful to learn about them by means of exploration environments. However it turned out that exploration environments were too difficult to use for users who had little experiences with computers or who felt insecure in handling computers (cf. Wulf 2000). These results indicate that exploration environments are obviously best suited for those users who have already reached a certain level of skills in handling computers. In the field study this feature was very well received by a local expert who tailored search tools for his colleagues. He used these environments to test newly tailored search tools.

Exploration environments, as proposed in this paper, allow individual users to experiment with groupware functions. So, they do not need to ask other users for access to their interfaces. Assuming that the learning styles of individual users differ considerably, we believe that this will lead to less mutual disturbances. On the other hand, in this way occasions for collective learning processes are diminished. These processes are nevertheless crucial when users learn tailorable applications (cf. Mackay 1990, Nardi 1993, Trigg and Bødker 1994). Therefore, the effects of these exploration environments on collective learning have to be investigated.

However, due to the fact that exploration environments are realized in groupware, the multi-user character of these applications can be used to support cooperative learning. For instance, users could be made aware of other users who experiment with the same function. So, they could either meet face-to-face or exchange their experiences via application immanent communication channels. In this way exploration environments could become virtual spaces in which users with a similar learning focus could meet. Moreover, one could represent local experts or support staff within these virtual spaces by means of their names, addresses or photos. So these actors could be contacted if other users need their support.

## 8. Conclusion

Exploration environments offer new possibilities to perceive the states transitions following the execution of groupware functions. In this way users can learn how groupware functions work. The evaluation study indicates that exploration environments increase the efficiency of computer supported task performance. Finally these environments could become spaces to build up learning communities around a groupware's use. This makes them an important feature for the design of any groupware.

Though this concept is applicable to a wide range of groupware functions, some preconditions have to be met. First, the behavior of the function to be explored needs to be deterministic. This precondition is typically given within computer applications. Nevertheless, it would not be wise to develop an exploration environment for the video or audio transmitting function of a video conference, in case it uses the internet as a transport medium. As the audio and video transfer rate is not deterministic it does not make sense to simulate these functions. Second, the behavior of the functions to be explored should not depend on real-time input of different users. As the input interfaces of the different users can only be accessed sequentially, it is not possible to simulate the effects of synchronous users' inputs. For instance, locking functions of synchronous group-editors cannot be explored in the way proposed here. Third, only those groupware applications can be equipped with exploration environments whose input/output interface allows simulating the input and output of other users. While this requirement is typically given with computer screens and keyboards, it often is not met with hardware-dominated output interfaces like telephone or telefax.

If these preconditions are not given, it will be not possible to realize exploration environment. In many other cases we believe that the concept of exploration environments will become an important means to support users in learning about groupware functions in general and tailorable ones in particular.

The concept of exploration environments, as it has been presented in this paper, can play the role of a general design requirement for groupware. With regard to the design principles of the ISO 9241 part 10, exploration environments can be seen as a potential means to realize suitability for learning. However, the concrete implementation of exploration environments requires the involvement of users. In the case reported here, the blinking arrow guiding the users' activities within the exploration environment was invented during usability tests of early prototypes. It turned out that this arrow had a positive effect on the users ability to navigate within the exploration environment.

The results presented here show the potential of exploration environments. However, we need more research results along the lines discussed before. Specifically, we need to develop further design alternatives and to get a better empirical understanding of its impact on users' learning.

## 9. References

Bentley, R.; Dourish, P.: Medium versus Mechanism. Supporting Collaboration Through Customisation. In: Marmolin, H.; Sundblad, Y.; Schmidt, K. (Hrsg.), Proceedings of the Fourth European Conference on Computer Supported Cooperative Work - ECSCW '95, Kluwer, pp. 133-148

Bortz, J.: Statistik für Sozialwissenschaftler, 4th edition, Springer, Heidelberg 1993

Dutke, S.: Mentale Modelle: Konstrukte des Wissens und Verstehens, VAP, Göttingen 1994

Carroll, J. M.; Mack, R. L.: Actively learning to use a word processor, in: Cooper, W. E. (ed.): Cognitive aspects of skilled typewriting, Springer: New York, 1983, pp. 259 – 281

Carroll, J. M.; Mack, R. L.: Learning to use a word processor by doing, by thinking, by knowing, in: Thomas, J. C.; Schneider, M. L. (eds): Human factors in computer systems, Ablex: Norwood, NJ 1984, pp. 13 – 51

Frese, M.; Albrecht, K.; Altmann, A.; Lang, J.; Papstein, P. V.; Peyerl, R.; Prümper, J.; Schulte-Göcking, H.; Wankmüller, I.; Wendel, R.: The effects of an active development of the mental model in the training process: experiment al results in a word processing system, in: Behviour and Information Technology, Vol. 7, No. 3, 1988, pp. 295-304

Greif, S.; Janikowski, A.: Aktives Lernen durch systematische Fehlerexploration oder programmiertes Lernen duch Tutorials, in: Zeitschrift für Arbeits- und Organisationspsychologie, Vol. 3, 1987, pp. 94-99

Howes, A.; Paynes, S. J.: Supporting exploratory learning, in: Proceedings of INTERACT'90, North-Holland, Amsterdam, pp. 881 – 885

ISO 9241: Ergonomic requirements for office work with visual display terminals (VDTs) Part 10: Dialogue Principles

Kahler, H.; Stiemerling, O.; Wulf, V.; Hoepfner, J.: Gemeinsame Anpassung von Einzelplatzanwendungen, in: Proceedings der neunten GI-Fachtagung Software-Ergonomie´99, Teubner, Stuttgart 1999, pp. 183 – 194

Kahler, H.; Morch, A.; Stiemerling, O.; Wulf, V.: Introduction to the Special Issue on "Tailorable Systems and Cooperative Work", in: Computer Supported Cooperative Work: The Journal of Collaborative Computing (JCSCW), Vol. 9, No.1, 2000, pp. 1 - 4

Kamouri, A. L.; Kamouri, J.; Smith, K. H.: Training by exploration: facilitating the transfer of procedural knowledge through analogical reasoning, in: Journal of Man-Machine Studies, Vol. 24, 1986, pp. 171-192

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

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

Oberquelle, H. (1994): Situationsbedingte und benutzerorientierte Anpaßbarkeit von Groupware. in: Hartmann, A.; Hermann, Th.; Rohde, M.; Wulf, V. (Hrsg.), Menschengerechte Groupware - Software-ergonomische Gestaltung und partizipative Umsetzung. Teubner Stuttgart, pp. 31-50

Oppermann, R.; Simm, H.: Adaptability: User-Initiated Individualization, In: Oppermann, R. (ed.): Adaptive User Support – Ergonomic Design of Manually and Automatically Adaptable Software. Hillsdale, New Jersey 1994: Lawrence Erlbaum Ass.

Paul, H.: Exploratives Agieren, Peter Lang, Frankfurt/M 1994

Shneiderman, B.: Direct Manipulation: A Step beyond Programming Language, IEEE Computer, Vol. 16, No. 8, 1983, pp. 57 – 68

Schmidt, K. 1991, Riding a Tiger or Computer Supported Cooperative Work, in L. Bannon, M. Robinson and K. Schmidt (eds), Proceedings of the Second European Conference on Computer Supported Cooperative Work, (Kluwer, Dordrecht, NL), 1 – 16

Trigg, R. and Bødker, S. 1994: From Implementation to Design: Tailoring and the Emergence of Systematization in CSCW, in: Proceedings of CSCW´94, ACM-Press, New York, pp. 45 – 55

Wulf, V.: Konfliktmanagement bei Groupware, Vieweg, Braunschweig und Wiesbaden 1997

Wulf, V.: On Search for Tailoring Functions: Empirical Findings and Implications for Design, in: Proceedings of OZCHI ´99, November 28 - 30, Wagga Wagga, Australia, 1999a, pp. 77 - 85

Wulf, V.: „Why did that happen?" – Building Appropriate Mental Models on Groupware Functions, in: Bullinger, H.-J.; Ziegler, J.: Human-Computer Interaction: Communication, Cooperation and Application Design, Lawrence Erlbaum, Mahwah, 1999b, pp. 338 - 342

Wulf, V: „Let's see your Search-Tool!" - Collaborative use of Tailored Artifacts in Groupware, in: Proceedings of GROUP '99, ACM-Press, New York 1999c, pp. 51 - 60

Wulf, V.: Exploration Environments: Supporting Users to Learn Groupware Functions, accepted by: Interacting with Computers, 2000, Vol. 13, pp. 265 - 299

Yang, Y.: Current Approaches & new Guidelines for Undo-Support Design, in: Proceedings of INTERACT'90, North-Holland, Amsterdam, pp. 543 - 548