End-User Controlled Group Formation and Access Rights Management in a Shared Workspace System

Joerg M. Haake, Anja Haake, Till Schümmer, Mohamed Bourimi, Britta Landgraf
FernUniversität in Hagen
Informatikzentrum, Universitätsstrasse 1
58084 Hagen, Germany
Telephone +49 2331 987 327
joerg.haake@fernuni-hagen.de

Abstract: Group formation and access rights management become crucial issues when shared workspaces are used to support flexible, emerging group work. End-Users should be able to form groups and adapt access rights to changing groups and workspaces. Current shared workspace systems do not support this sufficiently. Our approach combines a room metaphor-based shared workspace system with the key-metaphor for facilitating both, end-user controlled flexible group formation and end-user controlled access rights management. An evaluation of this approach with 290 users indicates that end-users can form groups and manage the access rights of their shared spaces.

1. INTRODUCTION

Many forms of collaboration, such as peer-learning groups or innovation teams in virtual organizations, require dynamic group formation. In such teams, e.g., availability of group members or the needed expertise may change. Thus, team membership will change over time. Likewise, different forms of group formation must be supported ranging from (1) assignment to (2) invitation to (3) free enrollment to (4) confirmed enrollment (i.e., interested people request group membership which is granted by the group). We argue that in these situations only the users themselves can define and adapt their groups properly. Thus, end-user controlled group formation becomes crucial.

Distributed groups are often supported by a shared workspace system, such as BSCW [1][2][3] or TeamRooms [14]. These systems offer shared spaces, where a group may meet, communicate and access shared artifacts. However, these systems support only limited forms of group formation, such as external group administration [14] or invitation of members into a shared workspace [1]. Thus, more fluent and emerging forms of group formation by the users themselves are not possible.

Emerging forms of group formation pose another problem to a shared workspace system that provides each group with a dedicated shared space: the problem of adapting access permissions to shared spaces with changing membership. Again we argue that only the users of a shared space themselves can properly define the access permissions. Given a large number of groups in such a system, no central administrator may serve all change requests in due time. Rather, the users themselves must be able to understand and control the access permissions.
In summary, we found that end-users have to be able to
create groups dynamically without prior planning of a system administrator,
employ different forms of group formation,
shape their own workspaces,
interact within the workspaces in various ways, and
control access rights to these workspaces.

Current approaches do not sufficiently support these requirements from an end-users perspective (cf. Section 2).

Our approach combines a room metaphor-based shared workspace system with the key-metaphor for facilitating end-user controlled flexible group formation and for facilitating end-user controlled access rights management. We implemented this approach in CURE, a web-based shared workspace environment. While previous papers on CURE [9] [10] reported solely on end-user tailoring of room structure and content for collaborative learning, this paper presents for the first time CURE’s functionality for group formation and access control.

Section 2 reviews related work on group formation and access rights management. Section 3 presents our approach. Section 4 reports on the results of the first 120 days of using the approach with 290 users. Section 5 compares our approach to previous work, followed by conclusions and future work.

2. STATE OF THE ART

Work on group formation and access rights management can be found in several areas:

Some systems support group formation only. Examples include agent-based expert finders or peer-learner finders [18][20]. These systems store profile data on its users and enable users to find partners matching certain constraints on their profiles. The set of matching users may be used to compose a group, e.g. of peer learners working on a joint exercise.

Access rights management is an important issue of data management systems. DBMS and shared file systems provide elaborated approaches for defining access permissions and controlling data access by central administrators. Access control lists and role-based access control [15][16] are standard means of managing access permissions in these systems. Much research has focussed on making access control more understandable (and usable for end users). A good overview on existing approaches for access control in collaborative environments has been provided by Bullock and Benford [4]. The authors proposed the SPACE access control model that used the concept of boundaries to group objects and control access to these objects. While the concept of boundaries is very intuitive (and as [4] argues can be found in many approaches), we see a problem in understanding the distribution of permissions for end users. If we for example look at the SPACE model, access control is managed by access graphs that define numerically, whether a user can move from one boundary to another boundary. For us, such a technical description seems to be difficult to comprehend and define for end users. They have to abstract from the virtual environments (constructed by boundaries) and move to a numerical model that only considers the structure of the environment. All these approaches do not support simple end-user controlled definition of access permissions.

Other approaches combine group formation and access rights management. Examples are place-based shared workspace systems, such as TeamRooms, VITAL, MOOs, and BSCW.
TeamRooms [14] provides persistent places. Each place provides a variety of synchronous tools for collaboration, and it also contains the shared data produced by these tools. Places cannot be nested. Access to places can be constrained by access rights defined by an administrator. Rights control whether users may create rooms or may forward access permissions to places to other users. These access rights apply to a complete server or to a list of workspaces. Thus TeamRooms does not support end-users in managing access permissions on individual places.

VITAL [13] offers rooms for learning activities. Teachers create rooms and learning material. Rooms support access to shared material and provide means for communication and group awareness. Learners can enter rooms and learn together with other learners or teachers. Those users working synchronously in the same room are considered a group. Thus, in VITAL groups can only be formed informally as a result of entering the same room. VITAL does not support access permissions.

CoWeb [8] is a WIKI-based system [12] complementing existing face-to-face groups such as classes. CoWeb supports asynchronous collaboration through information distribution and access to shared material. However, support for group formation was not reported in [8].

A metaphor of locks has been used in MOOs, where interaction is organized in places (such as rooms)[11]. By default, these places are public. The owner of a place can make it private by adding a lock to the place. The lock can be opened for specific users (by the place owner) using ‘unlock’ commands. Thus, in such a MOO groups may be formed only by the owner by assigning users to the place. While this approach uses consistent metaphors, it is not flexible enough to support other user-controlled forms of group formation without place administrators.

BSCW [1][2][3] provides public shared spaces and shared workspaces to invited users. A complex role-based access control method is used to define the roles of users. A role has a name and defines what a user having this role can do with the object/workspace. BSCW distinguishes three types of roles: normal, inheritable, and fixed roles. Roles may be redefined in workspaces. Members of a workspace are automatically considered members of all contained sub workspaces. BSCW supports group formation via invitation: the workspace owner can send invitations by e-mail to users, which may complete their registration and become workspace members. In BSCW, users have their own workspaces. They can create new sub workspaces and may decide to share them with other users they invite. A workspace becomes a shared workspace either by inviting other users into it, or by declaring this workspace to be a sub workspace of another, already shared workspace. In the latter case, the users of the original shared workspace become also users of the now contained sub workspace.

In our university, BSCW is used to support joint group work and joint learning for distributed groups, e.g., in virtual seminars and practical lab courses for non computer science students (e.g., studying psychology). Students were trained in using BSCW following a tutorial [6] for non computer science users. Then, students were asked to form groups and create group workspaces. Our experience shows that these students never created their own workspaces, nor adapted their group composition or access permissions. In fact, the teachers had to pre-structure workspaces and assign students to groups. Interviews revealed that students had problems understanding the functionality for creating workspaces, inviting group members, and adapting access permissions. Rather than learning how to use a computer application, they wanted to focus on their task (i.e., the seminar, the tasks in the lab course).

In contrast to place-based systems, Wessner and Pfister [18] organize cooperative work as activities. [18] describes an approach that supports group formation for cooperative activities during a course. Such a cooperative activity is modeled as an IPoC (Intentional Point of Cooperation). Teachers specify IPoCs before a course is started. IPoCs are characterized by the type and number of collaborators, duration, material, tools and didactical method to be used (called a learning protocol). Based on this specification,
the approach supports two kinds of group formation: Firstly, tutors can manually assign learners to IPoCs. Secondly, the system can compute groups of learners fulfilling the IPoC characteristics. Learners then receive invitations and need to agree to participate. If learners reject an invitation, the system tries to find another matching learner. Access right management is not mentioned in [18].

Finally, systems such as Yahoo!Groups [19] allow users to create and join groups for communication purposes. When a group is created it must be located in a hierarchical index of groups (the Yahoo!Groups directory). While the creator can propose an initial location in the index, a central coordinator may propose to change the location. Hierarchical subdivision of groups into sub groups is not supported. All members of a group have access to the group resources. E.g., they receive mails sent to the group, and may send mails to the group. Thus, the group lacks different modes of individual involvement (i.e. access permissions). Groups are formed by users subscribing to a Yahoo!Group via the index or by invitation. All above approaches support only limited modes of group formation (such as invitation only or free access). No system supports the complete spectrum of group formation ranging from (1) assignment to (2) invitation to (3) free enrollment to (4) confirmed enrollment. Access rights management by lay users is insufficiently supported, either due to too complex role-models, access control parameters and user interfaces that lay users cannot easily understand, or due to insufficient functionality. In summary, we can state that current approaches do not sufficiently support dynamic group formation and appropriate adaptation of access permissions for shared workspaces by lay users.

3. APPROACH

In this section, we present a combination of two metaphors and a corresponding user interface that support end-users in forming groups and managing access permissions for their shared workspaces. We begin with the room concept as a metaphor for places for collaborating groups and then extend it with the virtual key concept as a metaphor for both, access permissions for places and membership in the group using the place. Then, we discuss how centralized and decentralized management of groups and access permissions can be facilitated with this approach.

3.1 Rooms as Metaphoric Places for Interaction

Room metaphors [7][13][14][17] have been widely used to structure collaboration. The room metaphor uses the room as the representation of a virtual place for collaboration. Rooms can contain documents and participants. Users, who are in the same room at the same time, can communicate by means of a communication channel that is automatically established between all users in the room. They can also perceive all documents that are in the room. Changes of these documents are visible to all members in the room.

Users can enter a room, whereby they can now access the room’s communication channels and may participate in collaborative activities. Users can also add documents or artifacts to rooms. This means that, e.g., if they add a new document, from then on it is accessible by all other users in the room. When users leave the room, the content stays there to allow users to come back later and continue their work on the room’s documents.

Figure 1 shows a typical room in CURE. It contains documents (in the example, a user reads the document “Planning Game” in the room “CoPE”) that can be edited by those users, who have sufficient edit rights. It provides two room-based communication channels: a mail box and a chat. Users can use the room-based e-mail to send a mail to the room. Users of the room (with communication rights) will receive this message.
By providing a plenary room, sharing and communication in a whole class or organization can be supported. By creating new rooms for sub groups and connecting those to the team’s room automatically, work and collaboration can be flexibly structured. However, in order to support evolving structures of group work, group formation, access permissions for rooms, the creation of new rooms and access to the material in the rooms must be managed by the users themselves.

3.2 The Metaphor of Virtual Keys

We apply the metaphor of virtual keys to the metaphor of rooms in order to provide group formation and access control to rooms. Each room is equipped with a lock, and users need a key matching this lock in order to get access to this room. To support public areas, which can be entered by all users, so-called public rooms provide a default key for every user.

![Figure 1: A room in CURE.](image)

A virtual key represents not only the permission to enter a room but also specifies what the key holder can do in the room. We identified three different classes of rights: **key-rights** defining what the user can do with the key, **room-rights** defining whether or not a user can enter a room or change the room structure, and **interaction-rights** specifying what the user can do in the room.

Figure 2 shows which rights are currently supported in CURE. Users can select their desired rights for each class of rights. Rights are ordered from left (most restrictive) to right (less restrictive). A less
restrictive right always includes all more restrictive rights. E.g., if a user has the right to pass on a key, he also has the right to delete or return the key.

Figure 3 shows the property page of a room. Among other properties, it shows all users, who have a key for this room. The key is shown next to the user. To inform users about their permissions in this room, the rights linked to an actual key must be displayed. Two alternatives for displaying this information are (1) an enumeration of textual labels for the rights in each class, or (2) graphical encoding of rights as colors/icons. While the first alternative would repeat the textual labels for all keys distributed for the room and thus would occupy much space in the user interface, the second approach requires careful design facilitating easy learning and use.

We chose from these alternatives and combined them where appropriate. A color-coding adhering to cultural standards in the western cultures is associated to rights: green symbolizes that the user has all possible rights of a class while red stands for no rights. Intermediate colors (light-green, yellow and orange) signal rights in between these extremes. The meaning of the colors is used consistently in the system and combined with textual labels and graphical representations where needed:

In the context of the definition of rights associated with a key, both during key requests and key distributions, the background of the cells representing available rights is colored appropriately (cf. Figure 2).

If operations associated to a right are offered at the user interface, the respective button is colored appropriately and in addition equipped with the first letter of the label of the respective right. For example in Figure 3, the user inspecting the property page of the room is able to perform the operations of deleting his key (yellow button labeled D), returning his key (orange button labeled R), and entering the room (orange button labeled E). Furthermore, a bubble help displays the textual label of the right, if the user holds the mouse over a colored button.

If keys are shown for information purposes only, e.g., as the keys shown next to each user of the room in Figure 3 or in the room directory shown in Figure 4, we use a graphical representation using these colors. Each key is presented as a key-icon that has three segments, one for each class of rights. The user “Lukosch” for instance has a key with a yellow segment for the key-right (delete the key), an orange segment for the room-right (enter the room) and a green segment for the interaction-right (edit pages).
3.3 Group Formation using Virtual Keys

Thus, in CURE a group is defined as the set of people using a certain place, i.e. the group of people using a certain room. Consequently, at this time CURE does not support groups as first class objects. Consider the example of student working groups formed during a lab course in software development. Figure 3 shows one of these working groups consisting of eight people, which use the room “No Complaint”. This group of eight users may be considered being formed of two sub groups: Five users have full rights (totally green keys) whereas three other users (Bourimi, Lukosch and Schuemmer) are only allowed to delete or return their key, to enter the room, and to edit its content. They are not allowed to create more keys or to create adjacent rooms. The latter three users were actually tutors advising the first group of five students. The three tutors were invited into the room by the five students owning this place.
In self organizing virtual teams, the group formation has to be administered by the potential group members themselves. In CURE, this is achieved by letting the users distribute keys for rooms among themselves. The possibilities for group formation are implied by the operations defined on keys and offered to the users in a suitable manner at the user interfaces.

In a shared workspace system using rooms and keys, users enter the system via a public entry hall. Figure 5 shows a section of the overall room map. This map shows all existing rooms. The public entry hall is indicated by the top right box of the room map entitled “Hall”. Rooms adjacent to the hall, e.g. the room entitled “Allgemeine B…” in the room map, are positioned along the vertical hallway below the “Hall”. Rooms adjacent to “Allgemeine B…”, such as “Kurs 500: Pr..” are positioned along another hallway located left to “Allgemeine B…”. Another means of orientation is the room directory. Figure 4 shows the section of the room directory displaying the rooms of a project-based course in computer science on cooperative computer games entitled “Fachpraktikum CSCW: Kooperative Spiele”.

Starting from the hall, users can either navigate to accessible rooms (i.e., other public rooms or rooms for which they hold a key) or create a new room. If a new room is created, the user automatically becomes the room owner and the holder of the master key (cf. the green key behind the room “Leitung” in fig. 4). This key permits every possible operation on its room. The room owner can then create copies of the master key with potentially restricted rights (cf. Figure 2) and make them available to other users. Three distribution schemes are supported: Firstly, the room owner may specify a default key every registered user gets assigned when entering this room. Secondly, the room owner may explicitly assign keys to registered users. Thirdly, the owner may put keys on the door of the room. In contrast to key distribution, users may also actively ask for keys.

Different uses of replicable keys allow different modes of group formation: by assignment, by invitation, by free enrollment (placing the keys on the room’s door), and by confirmed enrollment (requesting a key from the owner). In addition, public rooms may be entered by every user registered in the system. In the following, we discuss each mode of group formation in more detail.

To make a room public, the owner of the room specifies a default key every registered user gets when entering this room. The default key defines the minimal access permissions (cf. section 3.4) of registered users in this public room. When a user is entering a public room and owns no special key for this public room, the system automatically assigns the default key to him.

For groups formed by assignment, the creator of the room determines the group’s members. In this case, she assigns other users to the group by creating copies of her replicable key that, among other possible restrictions, at least have no key-rights linked to the key copy. One of those keys is then passed to each user, who should be in the group. Receivers of keys are notified by E-Mail. Since in group formation by assignment the users receive keys with no key-rights, they cannot delete, return or pass on the key. This group formation scheme is, e.g., applicable to learning situations, where tutors have to assign students to groups working on collaborative exercises.

In order to form a group by invitation, the creator of the room creates and assigns keys with the key-right to return the key. Thus, users receiving the key may access the room and belong to the group, or they may reject the invitation by returning the key. Students inviting peer learners into a room for exam preparation may use this group formation method.
In case of a **group with free enrollment**, the creator of the room creates a set of copies of her key (which are usually copy protected) and puts them on the room’s door (outside the restricted area). The availability of free keys is visualized as a green door knocker, symbolized by a green ring in figure 4 (e.g. behind the room “Fachpraktikum CSCW...”). If a user wants to join the group, he clicks on the green door knocker, grabs a key from the door, and enters the room. The creator of the room can control the group’s size by putting only a limited number of keys on the door. If a user wants to join the group and all keys are already taken, he may subscribe to a waiting list for keys and will be notified when a key becomes available (e.g. new keys are put on the door, or a key is returned). One example for the use of keys for group formation by free enrollment is a virtual seminar: The teacher creates a lobby for the seminar and determines group members by assignment. For each topic of the seminar, the teacher creates a work room in the lobby room, and equips each work room with as many keys, as students should work on the topic. Students enroll in a work group on a topic by taking one of the keys of its work room.

In case, where the **enrollment in the group has to be confirmed by the room owner**, a user can join the group by knocking on the room by clicking on a yellow door knocker, symbolized by a yellow ring in figure 4 (e.g. behind the room “Multiplayer-Pacman”). The user can then specify the desired rights as shown in fig. 2 and explain why he should be part of the group. By submitting the form, the room owner will be notified by e-mail and can create an appropriate personal key for the user. This way, students could ask for access to tutor-guided closed learning groups.

If a user himself wants to leave a group, he can return the key to the room owner (or delete it, if he has sufficient key-rights). In case of groups with free enrollments, he may do so by putting his key back on the door. This clears his place in the group for another user. If the room owner wants to leave the group, she may destroy here master key if another user owns a master key, too. Otherwise, she is advised to pass her master key on to another user.

To grant group membership only for a certain time frame, keys may be equipped with temporal restrictions, e.g. being valid from a certain point in time until a certain point in time. To exclude users from a group, a key may also be retracted from a user by every other user of the group holding a master key (totally green key with all rights) for this room. As these mechanisms are primarily used for managing access permissions, we will discuss them in more detail in the next sub section.
In some cases, the room owner wants to make the room temporarily available to another group, too. Consider the case where in a virtual seminar during some period of time several related working groups should visit each other’s places. This would require assigning a new key to every member of the invited group, and would also require taking those keys back after the collaboration period is over. To simplify such temporary unions, an exchange lock operation is supported: Those users of a room possessing the “exchange room locks”-right may invite another group (i.e. the users of another room) by changing the lock of their room in a way that the keys of the invited users also match the modified lock.

As CURE focuses on evolving learning and working groups, groups are linked to evolving places (rooms). Thus, they are fluid and flexible teams focusing on the task to perform, instead of static organizational units. Our approach could also be used to support more static or organizational notions of groups. We could grant keys not only to individual users but also to organizational groups. Each member of such an organizational group could automatically get an appropriate key. The operations for moving and retracting keys could be easily extended to organizational groups. However, the current applications of CURE did not require this. Exchanging locks and thus joining groups as described above seemed to be sufficient at the moment.

In summary, CURE supports different forms of group formation. For an existing room, a user having sufficient key-rights to replicate and distribute keys may start group formation. Creating keys requires the copy-key right, and passing on keys requires the pass-key right. By using the above distribution schemes for keys, different forms of group formation can be implemented: from assignment to invitation to free access. In addition, the exchange lock operation supports the union of groups.

If a new room is needed, e.g. to work on a new task or to form a new group, the user needs in the current room the room-right of creating a new adjacent room. Once the new room is created, the above group formation mechanisms for existing rooms can be used. Thus, the available room-rights and key-rights constrain how groups can be formed in a CURE room.

Figure 5: A section of a room map giving users an overview of existing rooms.
In CURE, users are encouraged to build their own rooms, to structure their work, and to distribute keys from the first sessions on. CURE provides simple operations for passing and moving keys among the users. Keys are reflected appropriately at the user interface in all contexts where such operations are needed.

### 3.4 Managing Access Permissions using Virtual Keys

Rooms in CURE can contain sub rooms and content. Content is modeled as pages. Two types of pages are supported: content pages and binary pages. Content pages can be edited by a simple WIKI-like syntax [12]. The syntax empowers the users to include text, images, formulas, and links. Binary pages are used to upload and share binary attachments in the room.

We support two classes of access permissions for a room: Firstly, room-rights include the rights to enter the room, to create new sub rooms, to exchange the lock of the room, and to delete the room plus its content. Secondly, interaction-rights include the rights to read the room’s content (i.e. its pages), to use the room’s communication channels (e.g. mail, chat), to annotate content of the room, and to edit the room’s content.

Since group work is organized in rooms, access permissions are granted for rooms as a whole - not for individual pages inside a room. The access permissions a user owns for a certain room apply to all pages accessible in this room, independently of issues of ownership found in other systems (see Section 5). Thus, we made keys in CURE simple and omnipresent: you have some right and this applies to the current room. This simplifies access right management considerably. In our experience (see Section 4), this poses no problems for organizing group work.

The access permissions a user has for all pages contained in a certain room are defined by the union of the interaction-rights of all keys the user owns for this room. Since keys are omnipresent in the user interface and since menus interpret the access permissions and offer only possible operations, users are aware of their permissions and can make use of them on all pages in the room.

Initially, the access permissions for a room are defined by the room owner. But flexible emergent group work by definition constantly changes. Thus, access permissions must be adapted. We support two types of adaptations: spontaneous and pre-planned adaptations.

Firstly, room owners and all group members with sufficient rights may react to some new needs by changing the access permissions of the room’s users. They can simply retract some or all keys and assign new keys with changed access permissions (e.g. interaction-rights) to the users. Thus, ad hoc adaptation is possible.

Secondly, room owners may plan for certain phases of collaborative work. Consider the example of a working room of a sub group of a seminar. The students may all have the editing interaction right for the first eight weeks, which then, when the submission of the contribution is due, turns into the communication interaction right (to discuss the submitted papers). To implement such changing access permissions, the tutors of the seminar simply assign two keys to each student at the beginning of the seminar: one being valid from the starting date of the seminar lasting for eight weeks with the “edit”-right and another key being valid from eight weeks after the starting date of the seminar with the “communicate”-right (cf. figure 2). Thus, by distributing keys with limited validation period, different phases of collaboration can be supported.

In summary, by granting and retracting keys with different interaction-rights for a room and with different validation periods, access permissions can be easily adapted to the group’s needs. Since access
permissions are defined for rooms and their content as a whole, the granularity of access control is easier understood by users and usability increases. This is confirmed by our usage experience (see section 4).

4. EXPERIENCES

We used the system in three different courses in the winter term 2003/04: a virtual seminar in psychology, a one term project-based course in computer science and tutor-guided learning groups in a course on linear algebra. For these courses, the teachers provided an initial learning environment (a set of rooms) and managed the initial formation of the groups.

In addition to these settings, all students of the university could use the system in the context of other courses. Examples of groups that formed on their own are study groups for business administration or discussion groups for university politics. These groups did not receive any administrative help from the system administrators.

In this paper, we present our experiences regarding three aspects of our system:

Group formation: Were the users (i.e., teachers and students) able to form groups? Did they form groups in a flexible way (i.e. with changing group membership during the live cycle of a group)?

Group work: Were the users able to use the system to perform collaborative work in their groups?

Access rights management: Were the users able to manage the access permissions of their shared workspaces themselves?

Our experiences are based on the first 120 days (4 months) of system usage. We counted a total of 290 users. All user actions were logged in a log file, which contains information about each web request (each click) of the users. Since the basic system architecture is a web application, we can restore the full interaction from the log files.

In order to determine answers to above three questions, we combined log file analysis with user questionnaires. This combination reduces the risks of misinterpreting the logged interaction.

4.1 Experiences with Group Formation

Group formation by end users in our cases required the creation of rooms for each group and the distribution of keys to group members. Temporary groups would be indicated by rooms that were deleted, showing that the collaboration is finished. Dynamically changing groups would be reflected in changing group membership (e.g., users return keys, new users receive keys) or in the formation of sub groups (i.e. in newly created adjacent sub rooms). We therefore analyzed the log files for indications of such behaviours.

Before groups can form in CURE, corresponding rooms must be created. At the end of the 120 days, the system had 91 rooms. These rooms have been in use for an average period of 48 days. In total, the users created 257 rooms. The difference between created and deleted rooms indicates that the rooms were used for short term collaboration (and were removed after the group finished their work). 27 rooms were created by lay people. 19 of them were still in use at the time of writing this paper; the other eight rooms were removed by the lay people again. We conclude that teachers and students were able to create rooms and adopt the room structure to their evolving needs.

Users generated 1127 keys. They used different schemas (see 3.3) for composing groups using keys:
16 public rooms (i.e. the owner defines a default key every user gets when entering the room at the first time) were created.

429 keys were explicitly assigned to users, where the owner of a room knows the group of people, who should collaborate in the room.

590 keys were free keys attached to the door of the room. This method of key distribution was popular for situations, where access should not be limited, but the place owner wanted to stay aware of the group that could access the place. Free keys were taken by the users 381 times. 105 keys were returned after interacting in the room. Thus, we can conclude that users changed their group membership over time.

150 keys were explicitly requested by users from room owners. This reflects a situation, where a user wants to join an existing group. Most of these requests (136) came from lay people. 108 requests were answered positively. 88 key requests had to be answered by lay people. The average time from a key request to the first use of a key by the requesting user was 12 hours. Most requests were answered below the average time (median time 8.25 minutes).

The above numbers indicate that the users were able to fulfill the task of group formation. Groups were formed using different group formation methods. Room owners did create new keys for users, and users were able to request and receive them. We could also observe that additional keys were often created several days after the room creation and that keys were created while group work had already started. Thus, we can conclude that some users changed their group membership over time. This highlights the need for key distribution at runtime. Most administrative tasks were completed in a short time. From the log, we can also see that users used their keys to work in the desired rooms.

These interpretations were confirmed by a questionnaire we sent to a group of 28 computer science students at the end of a four month lab course in which they used CURE. During this period of time, the students were asked to develop a collaborative application. The lab course required the students to collaborate in groups. Each group used one group room in CURE as their main interaction space. At the beginning of the lab course, the students assigned themselves to groups. One student of each group created a room and the other students were assigned keys for this room by the person. All groups invited at least one of the tutors to their group room (by passing a key to this user). Several groups created an additional group room, in which private communication took place (without access by tutors). The questionnaire data from the 22 questionnaires submitted by students confirms that students had no problems in forming groups and inviting new members.

In summary, we found that users (i.e., teachers and students) in our study were able to form groups. They were able to form groups in a flexible way, as indicated by changing group membership.

4.2 Experiences with Group Work

From the log files we can see that users worked in those rooms for which they got keys. The interaction within rooms consists of communication (e.g. via chat and room mail) and manipulation of shared pages.

Mail communication was the most prominent way of interaction in the groups. During the observed period, users exchanged 1819 e-mail messages. Chat communication was also used, but mainly in pre-planned situations. Especially the virtual seminar conducted long chat sessions. In total, we registered 6641 chat entries. Users created 1031 plain text pages (the WIKI-like pages that are stored and edited directly in the room). In average, each page was edited 4.3 times. This shows that users refined the content after they initially created the pages.

The data from the log files suggests that group work happened in the group rooms. This interpretation is supported from the questionnaire data (see above). These students used CURE mainly for communication
purposes (each group exchanged approx. 250 e-mail messages by means of their CURE group room). Shared pages were used less frequently in the lab course, since the students used a separate version control system (CVS) to manage access to shared artefacts (e.g., documentation, code). However, the students reported that they used CURE for coordinating their software development project. Thus, we can conclude that the users in our study were able to use the system to perform collaborative work in their groups. Positive comments from students about the usefulness of CURE indicate the students’ positive attitude towards CURE.

4.3 Experiences with Access Rights Management

Access rights management in CURE occurs at two points in time: upon creation of the room, and when later adaptation is needed. When a room is created, the owner creates keys for the users of the room. By defining the corresponding key rights, the owner determines the initial access permissions of the (later) key holders. From the log files we can see that in most cases lay users (i.e. teachers, students) created keys with full permissions. This enables users to implement simple collaboration scenarios, e.g. based on mutual trust. Users used this strategy without reporting any problems. Some groups did show more elaborate use of key rights, such as some groups in the lab course that invited tutors into their rooms, but giving them only restricted permissions. The questionnaire data (see above) also indicates that students had no problems in setting up the access permissions for their shared spaces. They liked the simple set-up of rooms and the easy distribution of keys.

In our log files, we did not find evidence for adaptation of access permissions via the retract-key operation or via using time-limited keys. This led us to a closer look at the learning scenarios employed by the three courses. None of the didactical approaches used in these courses required users to change access permissions at a later stage. Consequently, these operations were not needed.

This finding surprised us, as some of this functionality was explicitly named by teaching staff of the three courses. Why did they not adapt their didactical strategy to employ the new functionality? An explanation could be that teachers may tend to replicate proven ways of teaching, especially under time pressure. When CURE was available for preparing the courses, not much time for experimentation was available. Interviews with teachers confirmed both, their continuing interest in more self-organized forms of learning as well as problems in adjusting the way of teaching to new technology. Thus, we are currently working with teachers on preparing courses employing new didactical approaches, which are based on more flexible group formation and access permission management.

In summary, we can conclude that users in our study were able to manage the access permissions of their shared workspaces themselves.

4.4 Discussion

As shown by our experiences, lay users are able to use CURE to form groups. The majority of groups remained stable, but some groups used flexible group formation. Users in CURE were able to perform collaborative work. They were able to manage the access permissions of their shared workspaces themselves. Users in general gave very positive feedback on the way rooms and keys are used in CURE to form groups and to manage access permissions.

These findings are based on data from 120 days of use with lay users. Our experiences show that the potential of CURE was not exhaustively used. We believe that this is a consequence of using traditional didactical approaches. New didactical approaches, which employ more flexible group formation and access rights management, may lead to increasing use of CURE’s more advanced features. Therefore, we
are currently collaborating with teachers on the development of such didactical approaches. The evaluation of the application of these approaches in CURE is our next goal.

5. COMPARISON TO RELATED WORK

Group formation in CURE is facilitated via rooms and the distribution of keys to group members. Different distribution/request schemas support different forms of group formation ranging from (1) assignment to (2) invitation to (3) free enrollment to (4) confirmed enrollment. By applying these schemas end-users can form groups as needed. Shared workspace systems such as VITAL [13] and CoWeb [8] do not support group formation and access rights management. Other approaches such as BSCW [1][2][3], TeamRooms [14], MOOs [11] and Yahoo!Groups [19] support only some of these forms of group formation.

BSCW offers so far the most extensive support for group formation and access rights management in shared workspaces: nested workspaces, group formation by invitation only and access rights management via roles. In the sequel, we focus the comparison of our approach to related work on BSCW.

In CURE and BSCW, a group is defined by the users of a room/workspace. BSCW groups may own a hierarchy of nested sub workspaces. If a workspace is included in a super workspace, the group members of the super workspace become automatically members of the new sub workspace. While this simplifies access right management in some cases, it may also cause problems if some sub groups should only have access to certain sub workspaces. This is, e.g., the case for virtual teams, which span company borders (e.g. sub contractors should only have access to some places in the project workspace hierarchy), or for learning groups that are dynamically divided into sub groups. CURE supports these situations by inviting the members of sub groups into the sub rooms only.

Access rights management by end-users is supported in CURE by the key-concept. A key defines three aspects of permissions in a room: key-rights, room-rights, and interaction rights. The owner of a room initially defines the keys for the later group members (users of the room). Later, the access permissions can be adapted by all users having the appropriate key-rights. They can retract and exchange keys, or they can define keys that are valid only during a certain time interval. BSCW uses roles to express permissions. They allow the propagation of access permissions including normal, inheritable and fixed roles. However, end-users then need to understand the underlying concept and propagation schemes. In addition, roles in BSCW with the same name may mean different things in different places. While this is a powerful and flexible concept, users must also deal with roles that may have similar names but mean different permissions in different workspaces. According to our experiences, the reduced complexity of access permissions in CURE is sufficient for the cases we encountered so far, and it facilitates end-user controlled access right management due to a simpler, more easy-to-learn concept.

One may argue that our approach of applying the access rights to a space’s content to all documents within a shared space uniformly is too simplistic and may cause the unintentional destruction of others work. To avoid or at least reduce such risks, BSCW support concepts like ownership of documents and soft locks on documents which influence and overlap with the effect of having a certain access right to a space.

Supposing that these interferences may also contribute to some of the confusion using these systems, we chose the WIKI approach [12] to prevent the accidental destruction of other peoples work: Whereas BSCW allows its users to explicitly enable version control on selected individual documents, CURE provides a build-in, all-time-active version control mechanism on all pages. Editing a document in CURE simultaneously, i.e. working in parallel on the same page, leads to parallel versions that can be merged later.
Moreover, in order to avoid unwanted versions, users may coordinate their behavior in different ways. If users are working synchronously, CURE provides group awareness by showing who is currently working in this room. Thus, users are aware of each other and may coordinate behavior using the chat tool of the room. Asynchronous collaboration may be coordinated via soft locks by putting comments in a content page or in the descriptor of a binary page. Since the first thing users do when beginning to work on a page is to read it, they will see the comment and may avoid conflicting access. However, individuals may still want to continue their work on the same page – and they can do this easily as their changes are automatically stored in another version of the page. As only the latest version of a page is shown in CURE (together with the user information and time stamp of the last update), users noticing changes have the option to consult the version history. Since old versions are easily accessible, a new version can be edited that merges different changes.

In summary, our approach goes beyond current approaches with respect to support for a large range of group formation schemas, and the provision of simple-to-use access rights management via keys.

6. CONCLUSIONS

In this paper we presented an approach supporting group formation and managing access rights in a web-based shared workspace system by end-users. Our approach is based on the metaphors of rooms and keys. Rooms provide shared workspaces. Keys define what access permissions the key holder has in a room. Groups are defined as the current key holders of a room. Group formation by end-users is supported via different schemas for key distribution/request. Access rights management by end-users is supported by operations (re)defining the set of keys of a room.

Our approach exceeds current approaches in group formation by supporting a range of forms of group formation: from assignment to invitation to free enrollment to confirmed enrollment. In addition, public rooms are also supported. The approach differs from role-based access control approaches by providing a simple room-related metaphor: keys.

Our experiences in the context of different forms of distributed collaborative learning in university courses show that end-users (lay users) are able to create shared spaces, form groups, perform collaborative work, and manage the access permissions of their shared spaces. Users gave very positive feedback on CURE. Although we could observe that users used not all functionality extensively, we believe that didactical approaches emphasizing more flexible group formation and access rights management will cause an increased use of this functionality as well.

Our next steps are to develop and test such didactical approaches with teachers and students at our university. We are currently making CURE an integral part of our university’s learning platform. From fall 2004 on, CURE will become a standard means for collaborative learning. This will provide us with more data on how CURE is used. CURE is available for public non commercial use at http://cure.pi6.fernuni-hagen.de.
7. REFERENCES


