Experience With Teaching and Learning in Cooperative Knowledge Areas

Thorsten Hampel  
Cooperative Media  
University of Paderborn  
Fürstenallee 11, 33102 Paderborn, Germany  
hampel@upb.de

Reinhard Keil-Slawik  
Computers and Society  
Heinz Nixdorf Institut  
Fürstenallee 11, 33102 Paderborn, Germany  
rks@upb.de

ABSTRACT
Since the early 1990s, different infrastructures for supporting learning have been developed and practically tested in Paderborn. One of the key products of our research is the web-based cooperative open source system sTeam (sTeam stands for Structuring Information in Teams). The sTeam platform is both a subject of research and an instrument for practical everyday use. Students and teachers meet in virtual knowledge areas where they can actively deposit, jointly view, exchange and arrange documents. In addition, they can annotate each other’s documents and link them to one another and to external resources. Authorization features such as user groups and access rights support such open and cooperative handling of material. This article reports our experience with the use of virtual cooperative knowledge areas in different courses held at the University of Paderborn. The focus is on mechanisms for the self-administration of cooperative knowledge areas by learners and on various techniques for organizing them to support learning processes in both physical classroom situations and distance learning. Key areas of research include techniques allowing learning material to be annotated and organized by learners as well as the electronic submission of completed assignments and the provision of course material in knowledge areas. It is important here to find a suitable compromise between necessary prestructuring and potential self-structuring of the knowledge space by the learners and to gradually accustom students to the new, unfamiliar scope offered by the web-based handling of material.

Brief Statement
This article makes an important contribution to research on the development of innovative web-based learning platforms and infrastructures featuring cooperative virtual knowledge areas. The open source open-sTeam platform is presented and its practical application evaluated.

Categories and Subject Descriptors
H.5.3 [Information Interfaces and Presentation]: Group and Organization Interfaces—Computer-supported cooperative work; Web-based interaction; H.5.4 [Information Interfaces and Presentation]: Hypertext/Hypermedia—Architectures; H.1.2 [Models and Principles]: User/Machine Systems—Human factors; K.3.1 [Computers and Education]: Computer Uses in Education—Collaborative learning; K.4.3 [Computers and Society]: Organizational Impacts—Computer-supported collaborative work

Copyright is held by the author/owner(s).

1. INTRODUCTION
The historical development of computer-supported tools for cooperative handling of work material (CSCW and CSCL) (cf. [3]) has been predominantly shaped by CSCW systems designed mainly for office applications. The focus here is largely on document management functions that allow documents to be organized and processed according to strictly defined rules and user rights arrangements. The most extreme example of this is, no doubt, so-called workflow management, in which even the path documents take between different cooperation partners and bodies is prescribed. Also, the ongoing integration of classical CSCW tools and web technologies has only slightly altered the basic paradigms of CSCW. Documents are made universally available, but user rights procedures and the organization of users in groups always follow fixed rules and laws prescribed by the specific application case. In the area of CSCL, there are a whole series of innovative approaches and tools. Some of these combine self-regulated/administered learning approaches with learning progress monitoring concepts or fixed learning-document flow schemas (cf. learning protocols, [11]). The drawback here is mostly that such tools are not web-integrated.

Given the fact that in recent years the World Wide Web has emerged as the de facto standard for publishing learning material, this is an unacceptable state of affairs.

Based on many years of experience with the development of infrastructures to support learning, we in Paderborn have for some years now been developing sTeam (Structuring Information in Teams) as an open source system for cooperative knowledge organization. The technical foundations of the sTeam system were presented in detail at WWW10 (cf. [7], [8]); the system integrates document management concepts with net-based virtual-community technologies (MUDs and MOOs). The presentation also dealt with the system’s theoretical foundations as a tool supporting primary media functions.

The present article begins by describing some of the salient features of the sTeam system in terms of the self-organization and self-administration of cooperative knowledge areas. These include, for example, innovative approaches incorporating the extensive use of annotations (cf. our implementation of the W3C Annotea ap-
proach in [6]) and mechanisms allowing completed assignments to be handed in electronically. It then goes on to address a second important area: our initial experience with the system’s use, particularly in terms of the self-organized structuring of knowledge spaces. Here, some initial fascinating results are available indicating that a culture for the self-organized structuring of knowledge resources and materials must first be developed for learners. sTeam is of special relevance to the web community because, on the one hand, it constitutes an everyday practical example of the use of web-based technology for cooperative knowledge management and, on the other, it represents an innovative framework for applications ranging from the design of cooperative teaching/learning environments to web content management systems.

2. DEVELOPING TECHNOLOGY TO SUPPORT COOPERATIVE LEARNING - THE OPEN SOURCE PROJECT OPEN-STEAM

To enable a large number of documents for a wide variety of courses to be managed and made available to students, a document management system has been in use by our group since the mid-1990s (Hyperwave, cf. [1], [10]). It provides asynchronous document management features, but is as yet, in terms of its basic architecture, not a cooperative system because it lacks, for example, mutual perception (awareness) mechanisms. Using specific architectural increments, the system was systematically extended to incorporate cooperative mechanisms and features for the long-term maintenance of digital teaching materials (cf. [2]). Parallel to this, innovative multimedia forms of knowledge presentation and interaction were designed and some of them were tested in interdisciplinary cooperation projects (cf. [5] and [9]).

Since 1998, sTeam has been developed as an open source platform for cooperative knowledge organization. Recent semesters have seen its first large-scale practical testing.

An important goal in setting up such an infrastructure to support learning is to enable aspects of cooperative learning to be studied in a practical, real-world context instead of under laboratory conditions. This is an essential prerequisite to obtain reliable results with respect to situated everyday activity (cf. Figure 6), leading to a productive refinement of the learning environment.

2.1 sTeam - an Open Source Platform for Knowledge Management and Collaborative Learning

A brief look at the potential uses of the sTeam system is unable to give a complete picture, first because of the broad range of possible applications, and second because this article is not meant to serve as a user manual. The idea is to present the basic concepts and give an initial impression of the potential uses of this cooperative knowledge organization system.

2.1.1 Frontends

To begin with, we must distinguish between the highly diverse frontends of the sTeam environment. Synchronous tools, such as the sTeam Shared Whiteboard (cf. Figure 2), allow a knowledge area to be jointly viewed. The user actions (mostly drawing and structuring functions or functions for manipulating the structure of documents) are represented such that they are synchronized among those participating in the knowledge area. Here it is a good idea, purely for the sake of clarity or the representability of actors’ actions using so-called telepointers, to confine use to small groups (2-8 persons) within one and the same knowledge area. Synchronous tools are thus suitable for cooperative situations of limited scope requiring a graphical drawing surface. Pilot projects using such sTeam access tools are currently being conducted in the Department of Design at Münster University of Applied Sciences. Here, a synchronous sTeam whiteboard is used to handle different
design processes, e.g. developing logos.

The application cases described here make exclusive use of the sTeam system’s currently most sophisticated web interface (cf. Figure 1). Here, a conventional web browser is sufficient to use the system’s asynchronous functions. We now go on to take a brief look at the potential offered by this web interface as an environment for cooperative knowledge organization.

2.1.2 Self-administration - Registration of Participants

The sTeam system is based on the idea of decentralized administration or self-organization of knowledge areas, which is why registration of new users is automated and net-based (cf. Figure 3). A clearly structured registration dialogue allows new users to access the system and request admission to a specific user group. Here, part of the password is sent as an e-mail to the new user in order to ensure the authenticity of the specified e-mail address.

Through their membership of different user groups, users obtain access to the system’s knowledge areas or are enabled to view and change the documents they contain, etc. Access rights are, then, assigned primarily on the basis of membership of different user groups. (User rights can, of course, just as well be linked directly to documents, but this is not usually the case.) User groups can be easily created. Users can, depending on access rights, create their own groups, taking into account different cooperation situations. Thus a student learning group needs both a group of its own and a corresponding area. New user groups are therefore automatically assigned their own areas. User groups also have a hierarchical structure, i.e. attributes of parent groups that are higher in the hierarchy are inherited by the child groups. This is a neat way of mapping admission and access structures to system areas and documents. Membership of different groups allows inheritance of access rights from higher-ranking to specific individual user groups. For instance, if a student is a member of a group in connection with attendance of a course of lectures, the access rights assigned to him or her through membership of this group (e.g. for annotating lecture material) are automatically inherited by the relevant tutorial or learning groups.

2.1.3 Areas/Web Interface

Our idea of cooperative knowledge organization revolves around the concept of the knowledge area (cf. Figure 4). In principle, all users can create areas, depending on their access rights. Such areas thus constitute the key element in any sort of cooperation between learners and the organization of material in a learning process. In the sTeam system’s web interface, areas can be represented as a space divided into four sections (cf. Figure 1). It should be taken into account here that the synchronous Shared-Whiteboard Client, for instance, can generate different views of one and the same area. Access tools thus generate views of areas depending on the possible ways of representing them or on their application environment. As shown in Figure 1, the web interface of an area representation can be divided into the awareness component (right), links (gates) to other areas (centre) and the documents the area contains (bottom). The essential elements of the cooperative knowledge areas metaphor are shown here in the form of semantic relations (gates and links) to other knowledge areas, in a user perception (awareness) component within a knowledge area as well as in the actual material of the knowledge area.

2.2 Documents/Material

The material of a knowledge area is basically composed of ordinary digital documents and graphics. These can be inserted into an area in different ways and displayed or loaded using an access tool (specific client or web interface). Within a knowledge area, documents can also be further organized by means of containers. In principle, containers are similar to areas in terms of their attributes, except for the fact that a user cannot move into a container and that a container cannot serve to restrict a synchronous communication such as a chat.
2.2.1 Rucksack/Inventory

Different functions can be used to manipulate or change the structure of the material in a knowledge area. A number of such functions can be encapsulated by the metaphor of a rucksack (inventory) carried by every user. For example, objects can be picked up and "dropped" again at various points. Similarly, copies and links (or gates) between objects can be created. For instance, first an exit to the area in which the user is currently located is created within the rucksack and the object (gate) can then be dropped at any desired location.

As pointed out above, access tools generate very different views of a knowledge area. Also, knowledge areas and containers can, depending on their function, have a highly specific representation - say, as a collection of images, e.g. lecture slides (gallery) or as a discussion forum. In all cases, views of different documents are processed with specific reference to the application context.

2.2.2 Annotations

A key cooperation instrument in asynchronous communication within the sTeam environment are annotations. With a view to linking communication as far as possible to the material of a learning process, each object can be annotated, depending on the relevant user rights. Readability of such annotations is based on the defined read access rights for different groups. Annotations to objects are seen as synonymous with the idea of news forums, so conventional newsreaders can be used to read annotations to material.

2.2.3 Rights and Groups

A central element in cooperative document management, and hence in an environment for cooperative knowledge organization like sTeam, is a hierarchy of user rights and user groups. Material can be assigned to groups of learners based on their membership of user groups. Here, the cooperation structure is basically determined by access rights for reading, writing, annotating and deleting documents. With a view to decoupling such rights from individual documents and specifically incorporating the cooperation context in a document's access rights structure, new forms of access rights inheritance are adopted in sTeam. A new feature here is the possibility of making rights dependent on an object's (document's) environment and specifically defining rights to pass on access authorizations (sanction right). The ability to define access rights in terms of context makes it possible to map necessary cooperative interactions among users. An example of this is the passing on of a document from one learner to another. Here, the rights are dynamically adapted to the document's environment. For instance, if the first learner has read and write access rights, these rights (e.g. a write access right) change in favour of the second learner if the document is passed on. Similarly, in the case of work assignments, handing in an assignment changes its access rights structure. While learners initially retain all rights to a document, once they have moved the document to, i.e. "dropped" it in, the virtual assignments submissions box, the rights are adapted in favour of the tutor. Now the assignment can no longer be altered by the learner, but it can be annotated by the corrector. Analogously, when a document is passed on from one learner to another, the rights are quite naturally adapted to suit the document's current environment, i.e. its context.

2.2.4 Self-administration - Decentralized Administration

Self-administration is another essential feature of the sTeam access rights system, in particular the possibility of passing on access rights. The aim is to enable decentralized administration and self-administration of cooperative knowledge spaces. Using simple mechanisms, learners register to participate in a group, e.g. a tutorial. For this purpose, there are group spaces analogous to user groups. Any user can be responsible for administering a group area, e.g. the tutorial group supervisor. Similarly, users can ask to participate in a tutorial group or can admit other users to their group.

The mechanism for passing on access rights enables areas and user groups to be centrally administered, depending on their respective functionality. If users create their own user groups with their corresponding areas, these can be administered with total autonomy. Here, the right to invite new users to join the area (to give new users access to the system) can also be assigned. In this way, one and the same sTeam server can handle highly diverse user groups and application cases. Each area can be administered centrally or self-administered by the users.

3. INITIAL EXPERIENCE WITH THE USE OF OPEN-STEAM

The system has been in practical use since the winter semester 2001/2002. The first course to be supported by the sTeam system was one regularly held by the Didactics of Computer Science group at the University of Paderborn. The course entitled "Grundlagen der Informatik für Lehrämter" (Foundations of Computer Science for Teachers) is offered as part of the further-training programme "Media and Information Technologies in Teaching and Education" and targets trainee teachers in all subjects. Here, the key concern of the working group was to assess the soundness of the virtual knowledge space concept in everyday use.

In the summer semester 2002, the system was used to support the course EIM - "Einführung in die Informatik für Medienwissenschaftler" (Introduction to Computer Science for Media Scientists), an interdepartmental course given by the Computer Science and Society group as part of the Media Sciences programme at the University of Paderborn. Here, the choice of the technology used was influenced by both the open nature of the user group and the experience gained with the system’s initial application. The system is currently in use on a larger scale in the course "Praxis der Systemgestaltung" (Practical System Design) with some 600 active users, as well as in smaller application contexts in Ulm and Munich.
3.1 Observing the System’s Use in the Course Foundations of Computer Science for Teachers (GIL)

In terms of its structure, GIL is a classical university course comprising lectures and tutorials. It is thus a course requiring students’ physical attendance, complemented by the net-based provision of course material and the organization of assignments using the open-sTeam system. The attendance requirement for a large part of the course meant that the system’s initial deployment was possible with only part of the functionality provided for in the concept. When the course was recently conducted, it had not yet proved possible to integrate a synchronous channel, e.g. a chat tool, in the browser-based user interface.

The use of sTeam basically involves two very different groups of users. The first group - and the larger one in terms of numbers - is that of the students. The second, though smaller, is more closely involved with the system: the group of the lecturers or, more precisely, the tutorial group supervisors. The course was not conducted by our own group but by the Didactics of Computer Science group, also at the University of Paderborn, which meant that they were relatively unfamiliar with the software used prior to deployment of the system. On the other hand, they were very keen to try out new teaching/learning concepts, which definitely had a positive effect on attitudes towards the technology used.

The group of students consisted of trainee teachers in a wide variety of subjects. This was due to the fact that the course was offered as part of an interdepartmental further-training programme in media skills for all trainee teachers, either during or after the second phase of their training for the senior section of secondary schools. This meant that participants were at an advanced stage in their studies, shortly before taking their examinations or in some cases having already taken their First State Examination. They were advanced students, then, with great variations in the level of their prior technical knowledge, possibly as a result of their differing specialties. In many cases, their technical skill levels must actually be regarded as low.

Ultimately, both groups of users had to be taken care of. As far as supervision of the tutors was concerned, the challenge was using open-sTeam for the very first time to support a university course. Together with the teaching assistant and lecturer of the Didactics group responsible for the course, a basic organizational structure was developed, which the participants were to refine to produce their own work environment. Contact with the Didactics group continued throughout the semester.

The students were supervised by means of participative observation. At the first joint session, the developers were presented to the course participants and their role and function were explained. At least one of the system developers was present at every tutorial session. Besides loosely participating in general discussions, the developer was available throughout to answer questions regarding the work environment.

The developers used this opportunity to observe the real procedures adopted by the users, and in some cases to clear up directly any misunderstandings concerning the use of the system. The students were given the chance to influence to a certain extent the further development of the system.

One demand on the part of the lecturers was that the course material should only be made available to the students within the system. Students were to be able to view the material at any time, but not to modify it. The tutor’s job, on the other hand, involved publishing the material, so the tutor had to be in possession of all access rights for this part of the course. The organization of the course was mapped to a group structure within the system in such a way that the course organizers formed their own group of GIL administrators, who in turn were members of the GIL group within the group of all course participants. This made it possible to take advantage of the fact that all the rights assigned to a normal course member were automatically transmitted to the group administrators/organizers. Conversely, further rights can be assigned to the group of GIL administrators.

The work area of the GIL group is used as the point of departure for organizing the material generated during the course. This seems a reasonable approach because admission to a group means that each member automatically gets an exit (gate) to the groups’ work area. This mechanism enabled accessibility of content to be ensured for the course participants. From here, the course structure branches into a lecture and a tutorial section. In the lecture section, a new subarea is created for each lecture session, in which the material provided by the lecturer for that session is deposited. In this section, then, we have an author-centred approach. The tutorial section is used in two ways: to publish the weekly assignment, and to organize the work for solving the problems in this assignment. A striking fact here is the gradual change in the means for submitting a solution as the complexity of the problems increases. While, initially, the assignments’ annotation options are used to provide a short answer, a special area is subsequently created for each tutorial session containing the submissions of all participants in the form of personal documents or groups of documents. In some cases, the spaces are even further subdivided into collections.

What is striking here is that a purely tree-like structure is created. This is all the more remarkable given the fact that the users tend to get annoyed about the long paths within the area structure. The length of these paths has to do with the fact that navigation within the tree always follows the tree’s edges. The structure does not contain transverse edges in the form of exits (gates), which could also have been inserted by students and which could be used to establish links, e.g. between the lecture and corresponding tutorial material.

None of the participants used the private work area, although this is one of the key concepts in the virtual knowledge spaces approach. The documents were placed directly in the group’s work area, sorted and then left untouched. Disappointing as this may seem at first sight, it does show, though, how important our basic assumption was that there must be areas within the system that can be jointly used by the group. In this case failure to use the private areas has presumably something to do with the type of projects being conducted. The solution of an assignment problem is typically completed in one step; it is not an evolutionary process in which the intermediate results are recorded for further processing. However, a user’s private work area basically serves as a universally

Figure 4: Virtual Knowledge Areas
accessible depository for his/her personal work. This function is assumed by the individual’s local workplace; the document is created, is by default not accessible to other group members, but can if necessary be explicitly made available to the other members of the course. Only if a product must be accessed from different points, and in several steps, is there a need for universal accessibility of a net-based private work area. The use of the annotation options within the open-sTeam system is also scenario-dependent. What is striking is that, as the complexity of the assignment problems increases, less use is made of the annotation option. This - at first sight - confusing observation is, however, not due to the fact that these problems require less discussion than the first, very simple problems, but to the fact that the annotations themselves were used to express a solution to the problems. As the problems’ complexity increases, the annotation option no longer appears adequate to describe the solution; instead, separate documents are created. Actual discussion of the assignment problems takes place, however, in the classroom, at the weekly tutorial sessions. This means that there is very little need for their discussion via the annotation option. If we focus our attention on when the solutions were placed in the system, we find that they were mostly completed shortly before the actual submission date for the assignments, leaving hardly any time for other participants to annotate them. One interesting observation is that in some cases the annotation option was used by participants to offer subsequent comments on their own solutions. Instead of the document being revised, additional comments were simply appended to it. Observation of the way the system was used during the course led to optimization of certain features of the user interface. For instance, the document annotation option - which was used extensively, especially in the early stages - was adapted to meet the students’ needs. The participants’ wishes included in particular support of simple marking/highlighting options within the annotation texts.

A special feature of the GIL course is the strongly hierarchical organization of the material. Students therefore complained about the long paths between the individual leaves of the tree. To shorten these paths, they were given the option of returning to any level of the tree in one step.

3.2 Supervision of an Elementary-Level Course - Introduction to Computer Science for Media Scientists (EIM)

Unlike the first pilot application, this course is for elementary-level students. It is normally taken by students at an early stage in their studies, i.e. generally in their second and at the latest in their fourth semester. Since it is classified as a compulsory elementary-level course, attendance figures are much higher than in the case of the small advanced-level course GIL. With over 100 participants divided into 4 tutorial groups, the organizational structure that has to be mapped in the system is much more extensive than in the previous use context. There are also changes in the way the assignments are processed. While in the advanced-level course GIL the focus was on communication about the assignment problems, which meant documents were handled openly, in the case of EIM there is a shift towards a more restrictive policy. The awarding of bonus points for the overall course examination, coupled with the requirement that assignments be completed, means that submission of the individual assignments acquires greater weight, affecting in turn the way the material is handled. In terms of open-sTeam’s access rights management, where the focus is on self-administration and maximum user autonomy, this leads to the challenge of mapping such structures.

3.3 Structure of the Course EIM

The tutorials were supervised by one of our co-workers (Joachim) and by a student teaching assistant (Silke). Prior to the course, neither of them had practical experience in using the system, though Joachim was familiar with the concepts of open-sTeam through discussions within our group. Before the system was deployed, then, both tutorial group supervisors were given an introductory briefing. Besides mastering document management functions for making available lecture and tutorial material, the focus here was on system structuring options.

Following our usual criteria for such courses, the course’s structure and size meant that the participants would be divided into four tutorial groups. Assignments are normally completed and handed in by small groups of two to three students. Practice has shown, though, that there is some cross-tutorial group building, so these small groups should not necessarily be seen as pure subgroups of the tutorial groups. It was therefore decided by the organizers to initially set up only the four tutorial groups. A special registration script was created, enabling students to register for one of the groups when generating their login on the open-sTeam server. This helped ensure even distribution of students across the tutorial groups because the script does not allow too many registrations for a particular group. The system’s administration tools do, however, allow subsequent re-allocation of individual students to other groups.

The building of small groups is left to the students themselves. They are told how to generate groups and how other students can become members of these groups. The naming of these groups and the organization of work on the assignments within the groups are also left to the students. This is where open-sTeam’s self-administration concept comes in. Apart from a minimal administrative core, in this case the setting up of the four tutorial groups, the supervisors have no further administrative tasks. The students set up their own user accounts and take care of their allocation to the tutorial groups themselves.

Students also assume responsibility for further organization of groups, the external requirement that a small group consist of two to three members being reflected in the actual group structure as it emerges. By the time the first tutorial sessions are over, each student is a member of at least three groups: via the registration script,

![Figure 5: Posting the Weekly Assignments](image)
in one of the four prescribed tutorial groups; through personal initiative, in one of the small groups; and also, indirectly through the subgroup relation of the four tutorial groups to a supergroup EIM, in this group which includes all course participants. The system’s automatic allocation to every member, on admission to a group, of an exit (gate) from the member’s home area to the group area ensures accessibility of lecture and tutorial material for the students.

The decision on the part of the course organizers to prescribe only the four tutorial groups and leave the other organization to the students established the basic structure of a group hierarchy. There remains the problem of how to implement within the system the official submission of the completed weekly assignments.

3.3.1 Weekly Assignments

To solve this problem, spaces are created which function like mailboxes. Here, the tutorial group supervisors have full access rights, while the tutorial members have only insertion (deposit) rights but no read or write access rights. This means that the tutorial participants can deposit documents in these spaces but can neither view nor read, let alone change, the submissions of other participants or their own submissions once they have been posted. Thus students posting assignments in these mailboxes do so more or less blindly. However, a notification mechanism was developed to provide the students with confirmation of the successful submission of their assignments. The documents are subsequently subject to the control of the tutorial group supervisors who correct and grade the assignments using the system’s integrated annotation mechanism to transmit their comments and evaluations to the students.

The returning of assignments is also performed within the system. No special mechanism was developed for this purpose; instead, the tutorial group supervisors used the platform’s standard mechanisms. The tutorial group supervisors arrange with the students to return the corrected assignments to the small-group spaces from which they were submitted. It is interesting to observe that two fundamentally different approaches were adopted here to reach the students’ small-group work spaces. Joachim created gates from the submission spaces of the tutorial groups supervised by him to the small groups involved. Silke, on the other hand, established a parallel structure by creating an area in her private work domain that she used like a corridor. Here, she created an gate for each small group, whose name contains the reference to the respective tutorial group.

3.3.2 Initial Results

In terms of the self-administration concept, this deployment of the system can be considered a complete success. At no point during the course, not even when setting up the initial administrative structure, was it necessary to resort to centralized administrative functions. From the perspective of the system, the tutorial group supervisors have the same status as any other participant. The limits of the self-administration concept became apparent whenever the tutorial group supervisors wished to assume administrative tasks for their students. In these cases, they had to resort to centralized administrative functions. One such case was where students had forgotten their passwords. In the sTeam system, only an administrator can issue a new password. As part of the current large-scale deployment of the system, a mechanism has been developed to automatically send a forgotten password to a previously specified e-mail address.

Also, the chosen procedure, whereby the tutorial group supervisors were to return the corrected assignments to the students’ small-group spaces, made it necessary for the tutorial supervisors to have insertion rights in the respective tutorial group areas. In principle, such rights could have been granted by the students themselves. To save time, however, in cases where students had not done this themselves, recourse was had to the services of an administrator.

The system’s functionality can be considered largely adequate in terms of supporting the course. No system adjustments were necessary, apart from the scripts for registering participants in the four tutorial groups and for supporting the submission of completed assignments. Here, the necessary adjustments were able to be made while the course was in progress.

The short initial training phase for the two tutorial group supervisors also sufficed to familiarize them with the system to the extent that they could both perform their own tasks and pass on their knowledge of the system to the students. Unlike in the GIL course, in the case of EIM there is no contact between the developers and the immediate users, the students.

Another positive feature of the system is that it evidently leaves enough scope to allow different users to solve one and the same problem in different ways, e.g. ensuring the tutorial group supervisors’ access to the small-group spaces.

4. CONCLUSIONS

Initial experience gained with the deployment of the open-sTeam system has shown: the use of mechanisms for web-based cooperative knowledge organization in teaching/learning environments requires a good deal more than simply providing a WWW-server. Besides questions relating to the establishment of a suitable infrastructure, particular attention must be given to user-oriented aspects, e.g. acceptance by the users of new scope in terms of system structurability.

The three-stage Paderborn Approach combines the theoretical media function concept (cf. [8], [4]) with the development of a concrete architecture and its evaluation in a practical academic context. Here, the theoretical media function concept offers the basis needed to enable technical problems to be distinguished from didactic problems and to define first of all the application-independent requirements with respect to material. Parallel to this, it is essential to establish a platform for web-based cooperative knowledge construction (cf. [7]) that meets the requirements in terms of free structurability and the arrangement of material but also allows self-organization of the learning environment. The result is the web-based client-server architecture sTeam with its highly flexible basic structure, which is pioneering especially in terms of its extensibil-
ity and adaptability to user needs. The product is not so much a concrete cooperative learning platform, but rather an open source platform capable of taking into account constantly changing needs, conditions and standards.

Despite the difficulties and obstacles encountered along the way, the two initial practical deployments of the open-sTeam platform can be considered successful. The platform has proved its stability in technical terms. And on the conceptual level, use was made of the essential functions and options offered by the system. For instance, students were able to register for the respective courses and enrol in corresponding user groups without outside help. It was possible to deposit and arrange various documents within individual or joint work areas, and mechanisms allowing the electronic submission of completed assignments were successfully tested.

However, what also became evident was that merely providing a specific functionality, e.g. the option of rearranging documents within work areas or creating new links/routes between work areas, does not necessarily mean this functionality will be widely used by the students - this would appear to require a culture of open cooperative work. And such a culture can only be created by long-term practice. Viewed pragmatically, open-sTeam thus exceeds the expectations and needs of some users. Looked at from the perspective of providing a flexible platform that takes into account differing use scenarios, functions such as self-administration are important and, indeed, indispensable. It is to be hoped that the decision, taken some three years ago, to establish sTeam as an open source platform will bear fruit in the form of a diverse and sustainable development.

5. ACKNOWLEDGMENTS

Our thanks go to Phil Bacon and Blazej Grabowski for translating and polishing up this paper. Special thanks also to all the sTeam system programmers, in particular Thomas Bopp, Bernd Eßmann, Daniel Büse and Ludger Merkens. This work was funded by the DFN-Verein as part of our open-sTeam Project.

6. REFERENCES