Hypermedia-based support for cooperative learning of process knowledge

Weigang Wang, Jörg M. Haake, Jessica Rubart and Daniel A. Tietze

IPSI—Information Publish and Systems Institute, GMD—German Research Center for Information Technology, Dolivostr. 15, 64293 Darmstadt, Germany;
E-mail: {wwang, haake, rubart, tietze}@darmstadt.gmd.de

The WWW makes learning materials widely accessible and provides an environment where people can learn across time and space. However, the simple read-only information structure on the Web provides little or very limited guidance to learners, especially when they want to learn how to do something through interaction with computers and other people. We suggest overcoming the problem by introducing a graphical hypermedia-based process representation and a cooperative process enactment support. The hypermedia-based process structure is accessible on the WWW. It contains a rich set of associated materials with which people can seamlessly interact while they are systematically carrying out the process they are learning under the guidance of computers or tutors. A hypermedia-based process support system and two use cases are presented. The use cases show that such a system can not only provide traditional learning support, but also offers novel cooperative hypermedia based support for the learning of process knowledge.

1. Introduction

In today’s global economy, organizations are becoming increasingly distributed. Furthermore, new forms of virtual organizations are emerging. As a consequence, teams carrying out business processes in a collaborative way are increasingly distributed and constantly changing. In addition, the business processes to be executed are changing in our global economy. Together, these trends require that such teams:

(1) learn what business processes they will execute collaboratively, and learn how they can execute these processes effectively in a distributed team;
(2) carry out the work collaboratively following the intended process; and
(3) adapt the process to the needs of the team and the actual situation.

Thus, a platform is needed which supports teams in the description and definition of processes, the learning of these processes, and the adaptation and execution of these processes.

The Internet and the WWW make information widely accessible and provide an environment within which people can learn and perform work whenever
they want and wherever they are. However, the simple read-only information structure of the Web creates several problems: Firstly, it provides learners with very limited guidance, especially when they want to learn how to do something through interaction with computers and other people. The simple HTML-based information structure can not represent a process structure graphically. Therefore, it is difficult for learners to get an overview of the process they are learning. Secondly, the read-only restriction prevents team members from changing a process description, either individually or cooperatively, based on the changing situation and what they have previously learnt. This restriction also prevents learners from getting interactive support from computers and other learners or tutors.

In this paper, we describe how to use and augment the WWW to overcome the above problems. Our approach to supporting teams in the description and definition of processes, the learning of these processes, and the adaptation and execution of these processes can be outlined as follows:

- use advanced hypermedia structures, i.e. extended links and composite nodes, to enrich the information structure of the current Web, as is required for process modelling;
- integrate process support into the advanced hypermedia structure that is superimposed on the current Web, as is required for supporting collaborative process execution;
- use the authoring and cooperation capabilities of a cooperative hypermedia system to support cooperative modelling, changing, and execution of both learning processes as well as the business processes to be learnt;
- make the cooperative hypermedia-based process support accessible on the Web; and
- provide traditional plus cooperative hypermedia-based support for the learning of process knowledge.

The remainder of the paper is organized as follows: In section 2, our enabling technology, the cooperative hypermedia-based approach to process support, is presented. Section 2.1 describes the principle of our flexible hypermedia-based process support. Section 2.2 discusses the shared workspace approach and how it is applied to hypermedia-based process support. Section 2.3 presents a cooperative process support system (XCHIPS) implemented with the approach. After these technical descriptions, section 3 uses two use case scenarios to describe how learning of process knowledge is supported by XCHIPS. For each use case, the use case scenario and our enriched support for the learning process are described. Section 4 discusses related work. The paper concludes with a summary and plans for future work. For a quick grasp of the cooperative learning aspects of the paper, readers may wish to go through section 3 before the more technically oriented section 2.
2. Cooperative hypermedia approach to process support

Our approach can be summarized by:

- modelling the process (i.e. the topic of learning) using hypermedia;
- integrating process support (i.e. support for definition and execution of processes) into the hypermedia model;
- providing cooperative hypermedia authoring and execution capability (i.e. a shared workspace) for working on the process description and executing it, and finally,
- providing cooperative learning support in this shared workspace.

In this section, the concepts of hypermedia-based process support and shared hypermedia workspaces and how they are combined in the cooperative hypermedia-based process support approach are presented. Then, a prototype system implemented with the cooperative hypermedia-based approach is described.

2.1 Flexible, hypermedia-based process support

A work procedure in a workflow system is defined by a workflow model composed of a set of discrete work steps with explicit specifications of how a unit of work flows among the different steps [1]. In general, a workflow coordination model can be defined as a directed graph, \((N, L)\), with a node set \(N\) representing individual steps in the procedure and an edge set \(L\) representing the coordination structure among the tasks [2].

The hypertext concept distinguishes information components (nodes) which are connected by relationships (links) [3], see Fig. 1. Using links, linear as well as non-linear network structures can be formed. In addition to the basic notion of nodes and links, one can introduce types of nodes and types of links (denoted in Fig. 1 by different darkness). These types can be used to capture application or domain semantics, e.g. by determining allowed types of nodes as link end-points of specific types of links. In addition to simple nodes, many hypertext systems introduced composite nodes (composites) that can contain other nodes and links (denoted in Fig. 1 by shaded rectangles and arrow lines). Thus, they can be used to form aggregated subnets within the hyperdocument, which lead to the possibility of layered graphs or networks. Hypermedia extends the hypertext concept by allowing any kind of multimedia information to be the content of nodes. Collaborative hypermedia now adds to the hypermedia concept the possibility of sharing a hypermedia workspace among many people.

Both the logical structures of a workflow process and a hypermedia network can be seen as a directed graph. This makes the correspondence between many workflow concepts and hypermedia concepts straightforward. For instance, tasks correspond to nodes. Control and data flow connectors between tasks correspond to links. Process execution corresponds to guided tours through a hypertext graph.
The key difference between these worlds is that process structures have their own computational semantics, such as task state transitions and flow dependencies, while hypermedia structures usually have no such semantics.

Our approach to modelling processes is to incorporate task-related attributes, such as state and time, into hypertext nodes, and incorporate control flow and data flow semantics into hypertext links [4,5]. Thus, a process is represented as a set of hypermedia task nodes connected by process links. Ordinary hypermedia links, nodes, and media objects are application-specific. They are usually not directly accessible to a workflow system. However, in this model they are represented in a unified hypertext model and therefore accessible by our system. This makes a hypermedia system not only a process support system, but also an information management system. Such a hypermedia-based process support system can provide support for a wide range of processes from very informal ones that guide people to perform their tasks to very formal workflow processes with automatic execution support. The composite structure of nested nodes of different types can provide different coordination support for differently structured subtasks. Such a hypermedia-based process description can contain a rich set of associated materials with which people can seamlessly interact while they are carrying out the primary task. More technical details on the cooperative hypermedia based process support can be found in [4,5, and 6].

2.2 Shared hypermedia workspaces

In order to support team members to cooperatively define and execute processes we suggest using the concept of a shared hypermedia workspace [7]. In such a workspace (see Fig. 2), users access shared and persistent hypermedia objects (i.e. nodes and links) either synchronously or asynchronously. Users interact with
these objects through tools (e.g. editors or browsers). Different node types can be viewed and edited using different, type-specific interaction tools. Such interactions may lead to changes of shared objects, which result in state changes of the entire workspace, which is in turn displayed to all users currently accessing the shared workspace. In this way, users can interact with each other through interactions with shared objects.

To support cooperation in such a shared workspace users need not only to access and manipulate shared objects but also need to communicate with each other and to coordinate their respective actions. Communication can take place outside the shared workspace (e.g. using e-mail, telephone or audio conferences) or inside the shared workspace, e.g. using a shared whiteboard node (a specific node type providing whiteboard-like contexts and for which a specific shared editing tool is available) or by exchanging annotations or notifications. Coordination can be supported by providing appropriate types of nodes and links in the hypermedia workspaces (see Fig. 3).

The content structure models the content of the shared workspace (i.e. the artifacts and their relationships) using typed hypermedia nodes (atomic and composite nodes) and hypermedia links. Different node and link types are supported, which can be used to express domain semantics as well as constraints. Using an extensible type system ensures that emerging task-oriented structures can be supported. For more details see [4,5].

The team structure models users and teams working in the shared workspace, again using hypermedia objects and representing relationships between users, teams and artifacts using typed hypermedia links. Examples for such relationships are teams being composed of users, users owning contents, and teams working on contents. This dual use of hypermedia supports simple editing and browsing of organizational as well as content structures in the global workspace. Note that
content and team structures are connected via relationships (such as, a user owns a node). These relationships can be represented as links.

The process structure models tasks and activities performed in the shared workspace. It uses nested hypermedia nodes (of type ‘task’) and represents dependencies (such as temporal order, control flow and data flow among tasks).
Hypermedia-based support for cooperative learning of process knowledge

between them using hypermedia links. Again, the process structure is connected with the content structure (e.g. a task uses a node as its work area) and the team structure (e.g. a task is performed by a team or a user).

In order to interact with such shared hypermedia workspaces three types of tools are provided:

- cooperative editors support individual and cooperative browsing and manipulation of nodes and links.
- A cooperative search tool supports queries on the shared workspace. In addition to the usual content-related queries, cooperation-related queries are also supported, such as:
  - team-related criteria: e.g. contents worked on by members of team X or by a specific user;
  - cooperation state-related criteria: e.g. contents being used in a specific cooperation mode or by groups of a certain size;
  - task-related criteria: e.g. tasks currently active, active tasks worked on by specific teams or users, contents being used as the local workspace of a specific task; and
  - change-related criteria: e.g. contents that were changed at all, changed during a specific time interval, changed by specific teams or users.

Since all the required information is present in the shared workspace representation, we implemented a shared workspace search tool that allows specification of the above predicates (for more details, see [7]).

- A cooperative navigation tool provides a hierarchical overview of the shared global workspace. In order to limit information overload, the navigation tool displays a hierarchy of nodes composed according to the containment relationship of composite nodes. This hierarchy is displayed as an indented outline view, which shows node titles, and which can be dynamically unfolded or collapsed. Additionally, collaboration-related information is displayed, such as the names of users and teams currently working on nodes in the workspace. Filters can be used to restrict the information displayed (e.g. to parts of the workspace, to certain users etc.). For more details see [7].

2.3 The XCHIPS system

Based on the cooperative hypermedia approach to process support, we have developed XCHIPS. XCHIPS stands for extensible Cooperative Hypermedia Integrated with Process Support. The XCHIPS system contains a set of tools (components) for people to create and use hypermedia objects cooperatively. It is implemented using Java and a Java-based Dynamic Collaboration Environment (DyCE) [8]. DyCE is a framework for creating mobile groupware components
that are loaded on demand. It provides dynamically replicated-shared data as well as transactional support for access to and modification of this shared data. Additionally, DyCE provides dynamic extension of the working environment by enabling users to add shared groupware components at runtime.

2.3.1 System architecture. The XCHIPS system has been developed using a Client/Server architecture (see Fig. 5). The server of the underlying framework (DyCE) provides persistence for shared objects and components as well as management services for both, e.g. uploading and downloading components, transactional support for access to and modification of shared data. The DyCE client provides a component desktop for accessing components registered to the logged on server. These components work on shared objects by using the counterparts of the object management services provided by the server on the client side. These counterparts communicate with the server services by using Java's Remote Method Invocation (RMI). In this way, DyCE provides a shared information space (or layer) which provides means for representing shared artifacts, for maintaining consistent replicas and for accessing shared objects.

The hypermedia workspace of XCHIPS is built on the shared information space. This hypermedia workspace contains hypermedia based information structures representing teams, processes, and the contents that the teams are working on. More details on the elements of the architecture are provided in the following sub-sections.

2.3.2 Components and shared objects. The XCHIPS system comprises components of each type of the shared hypermedia workspace tools (see section 2.2). Therefore, cooperative search and navigation tools as well as cooperative editors are provided. The XCHIPS collaborative components developed on DyCE can be

---

Figure 5. System architecture.
registered to a server and transferred across the Internet. These components may work in Java applications or Java applets that are executable in standard WWW browsers. In the XCHIPS system, both the hypermedia-based process structure and its associated information are shared objects represented in the shared data model and manipulated in mobile groupware components. The XCHIPS browser is one mobile groupware component besides lots of others (e.g. the search and navigate components or the shared notepad and whiteboard components for editing contents) that are used by it. These shared objects, as well as the components themselves, are stored on one or more DyCE servers and replicated to client applications when needed. The shared data can also be imported from and exported to documents in XML for exchanging data with external tools.

2.3.3 Process structure representation in hypermedia workspace. In XCHIPS, the process structure is represented by composite nodes and extended links. Extended links are links with more than one end point. Composite nodes are graphically presented as labelled icons and the extended links are graphically presented as labelled arrow lines between the node icons. The common WWW embedded links (i.e. the simple HTML links) can be used to point to the content information of a task or associated reference materials. They are presented as link markers with a special icon image. Composite nodes may contain other nodes, links, and information objects (URLs or arbitrary shared objects which DyCE components work on). An XCHIPS browser has been developed to present and edit the graphical process structure, its associated content objects and HTML links. The state information of task nodes is color-coded in their iconified views (white for inactive, yellow for enabled, green for active and brown for completed). Tasks are performed by menu operations activated on the task nodes. For the common WWW embedded links, when activated, their content pages will be presented in standard WWW browsers (i.e. Netscape and IE).

2.3.4 The XCHIPS browser and other tools. There are two object palettes on the XCHIPS browser: a tool palette on the left-hand side and an object palette on the right-hand side of the browser (see Fig. 6). The tool palette contains groupware tools and information organization tools that can be launched from the browsers. The groupware tools include a User Guide Browser, an Example Browser, a Paste tool, a Search tool, a Chat tool, and a Whiteboard tool. The information organization tools include hypermedia objects for creating Notepads, Embedded Links and Folders. The object palette on the right-hand side contains hypermedia objects used to represent workflow processes. These objects are ‘Start’, ‘Finish’, ‘Milestone’, ‘Task’, ‘Actor’, and ‘Meeting’ nodes, and ‘precede’, ‘coordinate’, ‘change’, ‘supervise’, ‘assign to’, and ‘participate’ links.

2.3.5 Sessions. The XCHIPS browser is itself a groupware tool (i.e. it can be used by multiple users at the same time). In addition, all tools activated from its tool palette or its content pane are also groupware tools. These groupware
tools are used in sessions. A session is a shared composite object formed of a collection of people, tools, and content objects. People can join a session to work together. Whenever a person joins a session, a session window will automatically be opened on his or her desktop. The groupware tools can be activated in the same session (thus becoming accessible to all users within this session) or in a new session (providing a simple transition between coupled cooperative and individual work). Users can invite other users to join their working session by activating the *Invite user* button at the upper right corner of the session browser. Users can also use the query tool to search for all the active sessions and request to join selected sessions.

2.3.6 Working modes. The XCHIPS browser can be launched in private, loosely coupled or tightly coupled working modes, and users may switch among these modes within a collaborative working process. In these working modes, some aspects of shared objects, such as the current node and the position of the scrollbar of the content pane, are coupled in different ways. For instance, in a loosely coupled cooperation mode, the navigation action of a user does not affect the browsers of other cooperating users. However, if users work in a tightly coupled cooperation mode, all users follow the navigational actions of other group members in the same session.
2.3.7 Integration with the Web. The integration of the XCHIPS system with WWW-based information spaces can take two forms: (1) use of XCHIPS in an Intranet or Internet Web site, and (2) access of information available through the WWW from XCHIPS workspaces.

For the first approach, we benefit from the fact that XCHIPS and the development framework on which it is based are developed in 100% pure Java. Therefore, the collaborative XCHIPS components can be embedded into Web pages to manipulate the shared artifacts (e.g. contents, teams, and processes) and to provide more background information about the shared artifacts. In this way, collaborative learning and training of process knowledge can be provided in corporate Intranets or via the Web-based information structures set up for distributed virtual organizations, providing almost ubiquitous access to shared process support.

The second approach makes use of XCHIPS to access ‘external’ information resources which are available from Web servers through a URL. In this way, the content structure part of the shared hypermedia workspace can be extended to encompass external Web references. Using XCHIPS’ support for ‘xref’ (see later section), elements of the XCHIPS workspace can be linked to external resources. During the course of the collaboration, these references can be activated to access the WWW resources. This access can either be through the regular Web browser or through a specific HTTP client component, also developed using DyCE, which then also allows collaborative Web browsing.

Using these two approaches, the collaborative hypermedia-based process support can be tightly integrated with Web-based information resources to extend the scope of the shared hypermedia workspace beyond what is available in XCHIPS itself. Using such extensions, background material for process learning or reference material for the context of the shared hypermedia workspace can be included in the collaborative process and the collaboratively created shared workspace contents can serve as meta-structure for HTML or XML pages.

3. Supporting cooperative learning of process knowledge in XCHIPS

In this section, we present two use case scenarios: first, the organization of a conference, and second, a work process in an extended enterprise. For both use cases, we first introduce the use case scenario. Then we use this scenario throughout all the subsections to show:

- how the process for learning how to do something (i.e. the learning process) and the process on doing something (i.e. the business process to be learnt) is supported, and
- how both traditional support and newer flexible, cooperative hypermedia-based support for the learning of the process knowledge is provided by the XCHIPS system.
Use case scenario: conference organization

Peter is a distinguished scientist. He is invited to serve as the next conference chair of a conference series. He has had no such experience before. Therefore, he wants to learn how to organize a conference, as conference chair. Peter’s friend John, who lives in another country, has served as conference chair for this conference series. Therefore, Peter contacts John for advice. Coincidentally, John has documented the process for organizing the conference in the XCHIPS system, so he simply sends a URL to Peter. The scenario is continued in the following sub sections in italic font.

Support for the learning process and the business process to be learnt. There are a number of different types of learning, such as learning to remember something, learning to understand something, learning a skill and learning how to do something. The learning process for learning how to do something is a simple process involving three steps [9]:

(1) determine the purpose,
(2) identify the procedures involved, and
(3) practice the task.

The following sub sections provide more details on these steps and describe how they can be supported by the XCHIPS system.

Determining the purpose A clear understanding of the purpose of what we are trying to learn may motivate us to learn. The XCHIPS system supports the explicit formulation of goals and allows learners to access this information to facilitate a better understanding. A hypermedia-based process structure may contain the process definition and other relevant information resources; the goals can be expressed as textual annotations to the process structure.

After receiving the URL from John, Peter types it into the URL field of the Web browser on his computer desktop. An XCHIPS browser appears and its content page contains a graphical representation of the process structure for organizing a conference (see Fig. 1). On the top of the content pane, there is a textual annotation that says, ‘This is a process for organizing a conference. The conference committee members are expected to understand and follow the process under the direction of a conference chair’. Peter is very happy, as this is just what he wants.

Identifying the procedures There are many ways to identify the procedures of how to do something. One way is to extract them from written instructions. Another way is to examine documented examples. The third way is to watch someone else doing it or interview someone who knows how to do it. The identified procedures on how to do something can be externalized in a graphical process representation, maybe first on paper and then in computer
Hypermedia-based support for cooperative learning of process knowledge

using the hypermedia-based process representation. In the XCHIPS system, the identified procedures on how to do something can be externalized in a graphical hypermedia-based process structure, in which tasks are represented as task nodes and precedence relationships between tasks are represented as process links. Other associated information, such as team- and role-based organization structure and referential materials can be represented using ordinary hypermedia nodes, links, and other content objects.

When John was to serve as a conference chair for the first time, he had the same learning need. He spent a lot of time reading texts on how to organize conferences, examining various conference Web pages, and talking to experienced people. When he finally figured out how to organize a conference, he documented the process in the XCHIPS system and used the process structure as a systematic guidance for learning and execution support.

3.1.1.3 Practising  We will learn how to do something successfully only by practising it. We can practise the tasks in a process following the identified procedures.

As described in the next section, Peter has studied the process structure and practised it virtually using both traditional learning support and cooperative hypermedia-based support provided by XCHIPS.

3.1.2 Traditional plus new kind of support. The headings of the following subsections are named after the widely used learning skills or teaching resources for learning how to do something [9]. In each subsection, we first briefly introduce one or two such traditional learning skills. Then, we describe how they are supported in the XCHIPS system. Finally, we continue the scenario to further highlight such support, especially the newer ingredients for cooperative learning.

3.1.2.1 Written instructions and examples  One way of learning how to do something is by following written instructions. These instructions summarize the main points to be learnt and provide us with background reference material. In the XCHIPS system, such instructions are provided as annotations and referential links at the side of the graphical process structure (see the ‘xref’ on the upper right corner of the content pane in Fig. 6. This ‘xref’ contains an URL, which points to a Web page describing the established regulations of the conference series). Additional information about how to use the process structure is provided in hyperlinked on-line documentation. This can also be referenced from within the hypermedia-based process structure.

Another way of learning how to do something is to examine and follow good examples. The hypermedia-based process structure can serve as an example. Learners can navigate through the process structure to get an overall picture of how to do something. They can also simulate the process execution to see how it proceeds when the actual execution starts.
Peter glances at the process structure and quickly gets an overview over how to organize a conference. He then has a look at the on-line help information of the system, and starts to navigate through the process structure to examine the content of the tasks and the sub-process structures, such as the sub-process for ‘develop conference program’. He also reads the instructions and reference materials attached to the process. After the first go, he returns to the root task node of the process structure, and triggers the process execution animation. The root task node first turns to ‘yellow’ (for ready), and then to ‘green’ (for active), then the process structure it contains appears. The start task in the process structure turns to ‘yellow’ and then to ‘green’, and finally to ‘brown’ (for completion). Then the tasks following the start task turn to ‘yellow’, ‘green’, and ‘brown’. The animation continues until all sub-tasks of the tasks are completed. Peter gets a feeling that a conference has been going smoothly towards its successful completion.

3.1.2.2 Demonstration and practice in a guided tour

One of the best ways of learning how to do something is to watch someone else doing it. A demonstration by a tutor is useful in that learners can see a model of correct behaviour provided by the tutor, which they can then imitate. With demonstrations, the tutor is also available to answer any questions that the learners might have.

Peter has some questions to ask. He invites John into his working session in the XCHIPS system. John volunteers to give a guided tour for Peter. As they work in a tightly coupled mode, the navigation performed by John also affects Peter’s browser. They use the Chat tool provided by the system in the same session as their XCHIPS browser, so that they can see both the process structure and the Chat window at the same time. John explains what he has done for each task during the tour and answers many questions asked by Peter.

3.1.2.3 Role-play and record-keeping

One of the most efficient ways of learning how to do something is to practise it. Role-play exercise is a situation in which one acts out, or performs certain skills and behaviours in a simulated situation. Each learner is given a certain role to perform and he or she is free to develop his or her role at wish, or according to guidelines that are provided. Role-play exercise is a valuable learning strategy in that it allows learners to practice new skills in a controlled, safe setting, before they are used in real environment.

Record-keeping of the role-play exercises should greatly improve our learning of skills because the records provide feedback of how we have performed and how others see us. Since all the information attached to a process is kept in the shared hypermedia space, newcomers can navigate the hyperspace to examine the process structures and the contents used or created in the process. Also, the process structure can be reused for performing similar tasks. In the XCHIPS system, the actors of a process take roles whose names indicate specific responsibilities. These roles are visualized and assigned to the tasks at the proper levels where they are involved in the nested process structure.
Hypermedia-based support for cooperative learning of process knowledge

Peter makes a clone of the process structure, does some more practice, and does some tailoring to his needs. He introduces the process structure and the XCHIPS system to his will-be program chair Tom. They work together on-line in the system, taking the conference chair and program chair roles. Peter works on many tasks, but only at a high level. Tom takes responsibility for developing a conference program. Peter also tries the Treasurer role. They edit the ‘execute conference program’ task jointly (Figure F shows an awareness indicator that Peter and Tom are working on this task). They split the task into two tasks: one for ‘Refine conference program’ and another for ‘Execute conference program’. They agree that it is a good idea to have the system used by all their conference and program committee members. Peter feels that he is well prepared to organize and chair the coming conference.

3.2 Use case scenario: extended enterprise process

An extended enterprise is dynamically composed of several partner companies who jointly work on a common project, and which may dissolve after the project is finished. In such an environment, a joint business process must be developed by the partners and must be performed by teams, which are formed from employees of the respective partners. Here, learning about the (just defined) business process is critical for the overall mission. This is demonstrated in the following scenario.

In order to develop a software product for a large customer, the Software Company X is creating an extended enterprise with two other partner companies Y and Z. Each partner has complementary expertise, skills and resources needed for the timely completion of the software product. Each of the partner companies is itself distributed over several sites, and together they form a large distributed virtual organization. In company X, Karl is the overall project leader for this software project. After his peers at Y and Z have been determined, Karl begins to work on the overall project plan in the XCHIPS system. After choosing a matching template from a library of project plans in XCHIPS and making some adjustments to their client he contacts his peers using the XCHIPS system.

What we will present next is how the team may carry out their learning task and may start to perform the project work in the XCHIPS system. The scenario is continued in the following sub sections in italic font.

3.2.1 Support for the learning process and the business process to be learnt. As explained in the first use case scenario, the team must determine the purpose of the business process, must identify the procedures involved, and must practise the task.

3.2.1.1 Determining the purpose Similar to the fist scenario, team members may individually or cooperatively browse the annotated process structure.

Karl and his peers agree to meet virtually in XCHIPS to examine the proposed project plan together. Karl explains the objectives of the project and navigates
the team to some background information, such as the preliminary requirement analysis and the contract with the client. Then, his peers start browsing through the project plan individually. After some time, they meet again at the top level of the project plan where they discuss open issues and record their resolutions as annotations of the project plan.

3.2.1.2 Identifying the procedures By examining the task decomposition hierarchy, Karl and his peers agree on the overall work distribution and how they will coordinate their respective tasks. Since each partner company has its own preferred business processes they agree that each partner will add descriptions of their processes as refinements of the top-level tasks. When they meet again, they walk each other through the added details, discuss alignments necessary because of different working styles, and record them as new annotations and dependencies in the joint project plan. As a result, they now know the big picture.

As indicated in the scenario, the shared project plan is used to negotiate common understanding among the project leaders. In the next phase, the employees of the partner companies need to learn about their new team players and their responsibilities.

3.2.1.3 Practicing After the respective project manager has decided staffing of the individual tasks, these assignments are recorded in the team structure and task assignment in XCHIPS.

Now, it is time for training the staff. Karl meets his project team in the XCHIPS system and introduces them to the proposed staffing. Since they all have browsed the shared project plan and learnt about the project objectives and their tasks, this is going smoothly. Each member of the project team is now meeting with his or her peers from the other partner companies. Each task manager already published date and time and location (i.e. task node) of these meetings in a shared whiteboard space. So, it is easy for the partners to meet virtually and to start practising (see next section).

3.2.2 Traditional plus new kind of support. As in the previous scenario, this setting also supports traditional learning steps in the collaborative environment. Again, the headings refer to the learning skills discussed previously.

3.2.2.1 Written instructions and examples As part of the shared project plan there are instructions, annotations and pointers to background information. In addition, simulations can be used to understand the sequencing of actions and their dependencies.

Hugo is task manager of the ‘develop repository’ sub-task. He has already selected some templates from project plans of previous successful projects and now starts to discuss them with his team composed of members from the different companies. After a previous in-person meeting his team mates already feel better about their new colleagues. By activating the simulation mode, Hugo demonstrates
to them how the project plan for their task could be executed. This sparks some discussions about the necessity of all steps and potential coordination problems. Some insight comes from seeing how intermediate results will be used later. Whenever needed, the group looks at the overall project plan to understand how their work is integrated into the overall project.

3.2.2.2 Demonstration and practice in a guided tour Since there are some open questions regarding the practicability of some actions, Hugo schedules a virtual meeting with an expert, Julia, who did this task in a previous project. Julia invites Hugo in a shared session. They use a search tool (see the upper left window in Fig. 7) to find the information related to the task in her previous project. The search result is displayed in a navigator window (see the lower part of Fig. 7). They also use a Chat tool (see the upper part of Fig. 7) to talk to each other. Together, they navigate through the different parts of the project to get an overview of the project in which the task that is similar to what Hugo will do is located. In Fig. 7, the user names at the side of ‘WP4’ provide a kind of

Figure 7. Query, navigator and chat components in one session.
group awareness information indicating that the two persons are working on (or browsing) the content of the task.

Hugo wants to make some annotations in Julia’s project. For that, they use the XCHIPS browser. However, before adding any annotations to the old project, they clone it and work on a copy. Hugo invites his team into the meeting. Julia walks the team through the project plan that was used in her project, points out difficulties encountered, and recommends some changes in the actual work plan of Hugo’s task ‘develop repository’ which they have opened additionally. As a result, some modifications are done and some experiences are recorded as annotations. The reference to the old project is kept, so that it can be revisited as a best practice example. In addition, team members can actually try out some actions under the guidance of the expert.

3.2.2.3 Role-play and record-keeping When it comes to actually performing the work, team members practise the task by taking appropriate roles in XCHIPS and enacting the project plan. The system then guides them through the process. Since in role-play mode users can simply simulate performing a task by activating the ‘complete task’ menu operation on the task node, they can explore what will happen. After some of these repeated role-plays the team now feels confident about performing the job, and real execution of the project model can begin in XCHIPS. Now, the learning process becomes a work process.

As seen in above scenario, learning and working are already interleaved activities. When team members browse through processes and try them out this may be called learning. However, when they detect problems in applying these processes to their task at hand, and they adapt the process structure to their needs, this may already be called ‘working’. When they finally feel confident and start performing the work in XCHIPS, they might view themselves primary in the mode of ‘working’. However, whenever they encounter problems and fix them by adapting the process structure (and recording their experiences) they fall back into ‘learning mode’. It is this ease of shifting between learning and working in an environment like XCHIPS that provides the most benefit.

4. Related work

Cooperative learning can improve several cognitive and social aspects of the learning process [10]. In a distributed cooperative learning setting, learners need some means to communicate and coordinate their learning activities. In the CSILE system, a certain cooperative learning method is modeled, and learners are guided through the process by specific rules [11]. Work in [10] and [12] provide some scripted learning support as learning protocols. The scripts are implemented procedures that can be executed automatically or on demand. The scripted learning support is a form of formal process support. Compared to scripted procedures, the hypermedia based process support provided by XCHIPS
is more flexible. The XCHIPS system can support a wide range of coordination means; from very informal shared artifact-based means to very formal processes that are suitable for different cooperative learning settings.

Process support is primarily provided by workflow systems. However, most workflow systems are not hypermedia-based and they often only provide users with a work list. It is not always easy to figure out the overview of a process structure and the current states of its constituent tasks. Some workflow systems also provide a graphical process editor. However, the editor is mainly for process designers, rather than end-users of the processes. The same is true for most scripting and agent-based workflow systems. Our hypermedia-based approach leads to a system dealing with processes and information structures (as the subject of processes) as two views on a unified data model. This permits smooth transitions between these two views, and supports cooperative work on defining and manipulating process and information structures. Unlike those workflow systems, which still maintain the strict separation between process and document data, the XCHIPS system can provide much more flexibility.

The user interface provided by most Web-based workflow systems uses simple HTML-based pages, in which only few components, such as Forms, provide some interaction support. Compared with HTML, Java is a fully-fledged programming language, its development kit provides a rich set of interface components for creating application-user interface. Therefore, we have implemented our user interface in Java. Java and XML have provided our system with a better accessibility and visualization support. In our approach, we still use HTML links to refer to our exported XML documents and other Web pages, and use Web browsers to invoke the XCHIPS system so that we can take advantage of the wide accessibility of the WWW.

A few previous hypertext systems have provided some form of process support. For example, Anecdote [13] provides support for a storyboarding process and the work in [14] provides an automaton-based hypertext for hypertext browsing and software engineering processes. However, their focus is not on cooperative modelling and execution of user-defined processes. The CHIPS system, a predecessor of XCHIPS, provides hypermedia-based support for cooperative process definition and execution [4,5, and 6]. However, in the past it had not been applied to the learning of process knowledge and, as with many systems in this category, it is not accessible on the web. Other XCHIPS extensions to CHIPS include the XML importing/exporting facilities and the dynamic software component extensibility, which allows the newly developed groupware components to be added to the system and made available to all its distributed users at run time. The advanced hypertext links and composition structure in XCHIPS enrich and complement the current Web information organization structure. They provide a process-enabled, graphically presented semantic network that is superimposed on web pages. Moreover, XCHIPS enables synchronous as well as asynchronous collaborative editing of shared processes and their associated information.
5. Conclusions and future work

In this paper, we described an approach that supports cooperative learning by providing:

- a shared hypermedia process model;
- integrated process support; and
- a cooperative environment for process definition and execution, which can be used to support cooperative learning as well as cooperatively performing the process.

In this approach, the cooperation environment is used as a medium for cooperative work and learning. Such a cooperative environment offers new possibilities for learning in distributed teams, such as animated simulations of processes, guided tours by experts through process descriptions and best practice examples, and cooperative role-play for practicing the task.

As a sample implementation of our approach, we described the XCHIPS system, which provides cooperative hypermedia-based process definition and enactment support for modelling, disseminating, and evolving process knowledge, and how it can be used for supporting cooperative learning of process knowledge on the WWW. XCHIPS presents the hypermedia-based process structure in a graphical form over the WWW or in stand-alone applications (using TCP/IP). In addition, the hypermedia-based process structure contains a rich set of associated materials with which people can interact while they are carrying out the business process they are learning. The processes can be executed automatically, under the control of computer, or manually, under the control of the process performers. The explicit process representation and enactment support can be used to guide people to practise and carry out the process they are learning. A prototype system based on the approach has been implemented. The two use cases presented in this paper show that the system can not only provide traditional learning support, but also offers novel cooperative hypermedia-based support for the learning of process knowledge.

As a next step in extending the XCHIPS/WWW integration, we intend to combine an XML import/export mechanism which is already included in XCHIPS with DyCE’s internal HTTP server. The resulting system will allow direct access, via a URL, to object structures included in the DyCE object space. In this way, any XML-compliant browser can be used to access XCHIPS object models. The shared hypermedia workspace and process support thus become extensive collaborative editing and management tools for information published on an Intranet or the Internet.

This paper focuses on the support for the learning of process knowledge in general, and the type of learning on how to doing something in particular. Our experience suggests that such cooperative hypermedia-based process support is also helpful for other types of learning, such as learning to memorize something
Hypermedia-based support for cooperative learning of process knowledge

and learning to understand something. The XCHIPS system is one of the tools we will bring into and extend in the EXTERNAL project (IST1999-10091) funded by the CEC. EXTERNAL focuses on the engineering and operation of networked organizations, and the management of process knowledge. Next, we will broaden and apply our approach to process knowledge management, integrate it with other EXTERNAL tools, apply it to other types of learning, and evaluate this approach in three real-world use cases. Furthermore, we are interested in combining the description of the ‘learning process’ (i.e. didactics) and the ‘process to be learnt’ (i.e. the content of learning) in a flexible manner. A second line of research relates to the combination of work and learning (as indicated in the second use case scenario). Here, learning smoothly turns into working (e.g. by adapting process structures to the task at hand) and vice versa (e.g. when a problem occurs during process execution and the team has to learn how to fix it, and to document it for future learners).

Acknowledgments

We thank Martin Wessner for his helpful comments on a previous version of this document.

 References

Weigang Wang is a research scientist at GMD-IPSI. He received his Master degree in computer science in 1988 from the Beijing Institute of Technology. He received his PhD in computer science in 1995 from the University of Liverpool. He has lead/is involved in the development of multiple co-operative hypermedia systems (MUCH, RICH, CHIPS and XCHIPS). Currently, he is the GMD site manager of a CEC project (EXTERNAL IST1999-10091) on engineering and operation of extended enterprises and process knowledge management. He has published over 40 refereed conference and journal papers. His main research and project areas are hypermedia, groupware, object-oriented modelling, flexible workflow, extended enterprise environments, and integrated cooperative working and learning environments.

Jörg M. Haake is manager of a research division at GMD-IPSI. Since 1989 he is conducting research in hypermedia and CSCW. Dr Haake received a MS degree in computer science from the University of Dortmund and a Doctoral Engineer degree in computer science from Darmstadt University of Technology. He is co-author of more than 40 refereed scientific publications and served on many program committees in the hypermedia, multimedia and CSCW fields. His current research interests include design, implementation and evaluation of open solutions for computer-supported cooperative work, cooperative learning, and their integration.

Jessica Rubart received her degree in computer science at the University of Paderborn (Dipl. Inform.). From 1994 to 1999 she worked for the Cooperative Computing and Communication Laboratory (C-LAB) in Paderborn as a student and later software engineer. Since 1999 she has been a computer scientist at GMD-IPSI in Darmstadt. Her main research and project areas are distributed cooperative working and learning environments, software engineering and construction tools as well as networked hypermedia.
Daniel A. Tietze studied computer science at the University of Bremen and has been working as a full-time researcher at GMD-IPSI since 1995. His research interests focus on computer support for tele-cooperation and tele-communication as well as object-oriented software engineering. His PhD work is concerned with the development of flexible and extensible groupware environments.
Author Queries
Msp. 13, (down) Which Section?
Msp. 18, (up) Figure F?
Ref. no. 9, Location of Publisher?
Ref. no. 8, Please update?