Organizing Shared Enterprise Workspaces Using Component-Based Cooperative Hypermedia

Jessica Rubart, Jörg M. Haake, Daniel A. Tietze, Weigang Wang
IPSI – Integrated Publication and Information Systems Institute
GMD – German National Research Center for Information Technology
Dolivostrasse 15, 64293 Darmstadt
{rubart, haake, tietze, wwang}@darmstadt.gmd.de

ABSTRACT
Cooperative work in Extended Enterprises needs a flexible shared workspace for team members to access and manipulate shared information objects in a well-coordinated working process. Current shared workspace systems do not adequately support the evolving character of shared workspaces as needed by Extended Enterprises, i.e. the dynamic cooperation processes, various kinds of shared information contents and the set of cooperative tools. In this paper, the usage scenarios and requirements developed in a European Extended Enterprise project are used to derive the requirements for shared enterprise workspaces.

Our approach utilizes component-based cooperative hypermedia to organize shared enterprise workspaces that contain team and process structures, information contents and their corresponding tools. The approach extends classical hypertext models to shared hypermedia objects as well as dynamic bindings between these and the Groupware Components working on them. To demonstrate the approach, a prototype system and a prototypical usage scenario are presented.

KEYWORDS: Shared hypermedia workspaces, cooperative hypermedia, software components, process support, extended enterprises, virtual enterprises

INTRODUCTION

Shared workspaces are very important for Computer Supported Cooperative Work (CSCW) applications since cooperative work requires information sharing. This paper deals with organizing shared enterprise workspaces in the sense of structuring information and supporting the cooperation processes. Hypermedia is a powerful concept for organizing information. Component-based systems enable end-users to compose components in a variety of ways and to add or remove components at any time, thus achieving extensibility and tailorability. Our approach combines both concepts – shared hypermedia and software components – and extends classical hypertext models like the Dexter hypertext reference model [12] and the core data model of the open hypermedia framework [20] to shared hypermedia objects. In addition, it provides dynamic bindings between shared hypermedia objects and Groupware Components that manipulate the hypermedia objects. In this way, a tighter integration of cooperation services, hypermedia objects and Groupware Components is achieved.

Our approach is motivated by the European project EXTERNAL ("EXTended Enterprise Resources, Network Architectures and Learning") [5] funded by the CEC (IST 1999-10091). There, we use shared workspaces for supporting dynamic networked organizations. Existing shared workspace systems do not adequately support the identified requirements. Thus, we have developed our component-based cooperative hypermedia approach.

Shared Workspaces

Shared workspaces aim at supporting cooperative tasks. They provide users with a virtual space in which information can be shared and exchanged [17]. Shared workspaces focus on cooperative information sharing in the sense of cooperative authoring, commenting, and annotating shared documents as a group activity [2]. They are used for long-term synchronous and asynchronous cooperation [11]. Synchronous cooperation means cooperation happening at the same time with typically fine-grained notifications giving immediate feedback about the activities of other users, whereas asynchronous cooperation can happen at different times with usually no notifications of other users' actions [13].

Dynamic Networked Organizations

Dynamic networked organizations are a new form of organizing business activities in our evolving global economy. Virtual Enterprises (VEs) and Extended Enterprises (EEs) are two examples [19, 8]. These forms of organizations are based on distributed cooperating teams. In the context of the EXTERNAL project an Extended Enterprise Network (EEN) is understood as an inter-organizational network, which provides a basis for partner enterprises to form VEs on demand. An Extended Enterprise (EE) consists of an EEN and of a set of VEs. VEs are dynamically composed of several partner enterprises, which work cooperatively on a common project. For this, the partner enterprises have to develop and perform a joint business process. Resources (artifacts and humans) have to be modeled and assigned to the different subtasks of the joint business process, and the joint business process must be performed cooperatively by cross-organizational teams. All elements of this cooperative setting (i.e. partners, work plan, cooperation processes, information structures and tools etc.) are highly variable. It is this dynamic nature, which makes supporting this cooperation so difficult. While a VE ceases to exist after the project is over, the EE continues to exist and the EEN provides a harbor of experience for forming new project-based VEs.

The remainder of this paper is organized as follows: The next section deals with requirements extracted from EXTERNAL. Then, our component-based cooperative hypermedia approach, a prototype system and a prototypical usage scenario are presented. Afterwards, we present related work regarding organizing shared workspaces. The paper ends with conclusions and plans for future work.

PROBLEM ANALYSIS

In EXTERNAL, five European organizations are involved in developing a methodology, an infrastructure, and tools needed to support EE engineering and operation. Three usage scenarios are used to validate the applicability of our approach: (1) supporting project planning and execution for software-oriented research and development projects, (2) supporting consulting in business development, and (3) supporting networks of small and medium enterprises in performing joint projects. In the following, we present technical requirements for the supporting technology, which are derived from the requirements analysis in EXTERNAL.

Within the requirements analysis of EXTERNAL [24] we have identified that all usage scenarios involve evolving cooperative work processes. Users of a shared workspace system work cooperatively on one or more tasks. By doing this, they produce documents, either by synchronous or by asynchronous cooperation. In the course of this paper, we use the term “document” to refer to a complex shared artifact, consisting of – potentially – several items. A shared document can be viewed and manipulated using interactive tools. These documents are put in the shared workspace and the resulting information structure should reflect the cooperation processes. Hence, a shared workspace system should support the definition of cooperative work processes, i.e. different tasks and subtasks assigned to resources. Therefore, we have identified the following requirement:

R1. Flexible ways for structuring the shared information space are needed to support the cooperation processes and the assignment of shared documents to tasks and to users, and of users to tasks.

Since cooperative processes evolve over time, shared workspaces constantly change, so that their organization should be adaptable at run-time:

R2. The organization task itself (i.e., the organization of the shared workspace and the definition of cooperative working processes) should be part of the shared workspace in the sense of being shared information that can be worked on by synchronous or asynchronous cooperation.

For explicit communication and discussions about their work, users of shared workspace systems need explicit communication support. Hence, the next requirement we have identified is the following:

R3. Means for communicating and discussing both, shared documents as well as shared process descriptions (composed of nested tasks) should be available.

Furthermore, users have to get an overview over which tasks have to be performed on which documents. Additionally, they should be aware of who is doing what. All EXTERNAL usage scenarios require supporting workspace awareness. This is meant to be an up-to-the-moment understanding of another person’s interaction in a shared workspace [10]. Workspace awareness improves the usability of cooperation:

R4. Workspace awareness should be supported to provide users with information about the current state of their cooperative work. Additionally, users require an overview over the entire shared workspace structure (including all users, tools, documents and processes).

For working on shared documents, users of a shared workspace system need tools. Using these tools, they work in synchronous or asynchronous cooperation mode on the shared documents. Every user has to have access to all necessary tools. The three usage scenarios of EXTERNAL require very different tools for working on shared documents. Additionally, new tools become necessary in an unforeseeable way. This leads us to our next requirement:

R5. The set of cooperative tools in the shared workspace should be easily extensible at run-time in order to use new cooperative tools for new or evolved tasks.

Finally, the EXTERNAL usage scenarios require easy accessibility of the shared workspace system. The World Wide Web is a common infrastructure that addresses interoperability with different computing platforms, is universally available, and requires no additional installation or maintenance except for the initial Web browser installation. Nowadays, this is part of every computing environment. Thus our last requirement is:

R6. The shared workspace system should be accessible over the World Wide Web and it should be possible to connect it with other web resources.

The next section describes our component-based cooperative hypermedia approach for addressing these requirements.

THE COMPONENT-BASED COOPERATIVE HYPERMEDIA APPROACH

When applying a hypermedia approach to shared enterprise workspaces, four main aspects arise from the problem analysis, which need to be addressed by a shared enterprise hypermedia workspace:

- Cooperative access to and manipulation of shared hypermedia structures, which are used to structure the workspace, must be supported to address R1 and R2.
- Cooperative modeling and execution of shared work processes must be supported in the shared workspace to address R2 and R4.
- Run-time extensibility of the hypermedia structures (i.e. new types of nodes/links, and new types of content) must be supported to address R1 and R6.
- Run-time extensibility of the tools/views operating on shared (emerging) hypermedia structures need to be supported to address R3, R5 and R6.

The first two aspects require generic cooperation support and emergent workflow modeling. Here, solutions have been presented, e.g. [11] or in CHIPS ([28], [29]).

The third and fourth aspects, though, pose new problems for hypermedia approaches. While Dexter-based and open hypermedia protocol-based systems have been designed to accommodate integration of hypermedia systems and exchangeability of hypermedia data, cooperation support was not a prime issue. In fact, although not explicitly stated, these approaches assumed a stable, pre-defined hypermedia structure (in terms of hypermedia object types and tools needed to work on them). Our problem analysis shows that this implicit assumption does not hold in EEs. Rather, hypermedia object types and the tools operating on them need to be modified at run-time. In the next subsections, we present our solutions to these problems.

Cooperative Access to Shared Hypermedia Structures Using Components

As an overview, in our approach, a shared enterprise workspace consists of a shared hypermedia object space as well as a shared component space (see Figure 1). The shared component space contains Groupware Components[1] working on shared objects and bound to shared object types, respectively, e.g. a cooperative hypermedia editor is bound to the composite type with the binding name "open". Users manipulate shared objects by using shared components in sessions. Hence, a session combines a set of users, Groupware Components, and the shared objects manipulated by the components. The shared object space is represented by a hypermedia structure, which also contains hypermedia object types for process support (see next subsection). The shared component space contains cooperative tools for manipulating, navigating and searching through the hypermedia workspace structure as well as for editing the actual contents. The component-based architecture provides easy deployment and extensibility of cooperative tools and shared object types as well as tailorability of the cooperation environment.

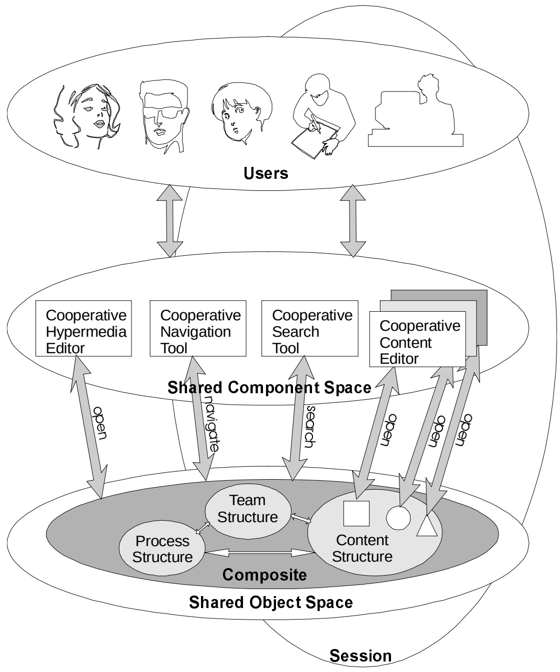

Figure 1: Component-Based Cooperative Hypermedia

By employing shared hypermedia objects as well as bindings between shared hypermedia objects and Groupware Components, a tight integration of synchronous and asynchronous cooperation services, hypermedia objects and Groupware Components is achieved.

Cooperative Modeling and Execution of Emergent Work Processes

Our model is based on experiences from our earlier work on CHIPS [28, 29]. Extensions to this work are motivated by the requirements of EXTERNAL. We model the work processes in the shared enterprise workspace using shared hypermedia with process support [28] (see Figure 2). Shared hypermedia with process support provides rich information structures, which reflect the semantics intended by users when creating both the information structure and the process description. The computer can interpret this information to provide cooperation support (as required by R1).

The basic hypertext concept is characterized by a node-link paradigm [16]. Nodes are connected by links, which allow linear as well as nonlinear network structures. Composite nodes can contain other nodes and links and therefore allow nested structures. In the Dexter model [12], nodes, links and composites are unified to the term "Components". In the core data model of the open hypermedia framework [20], a wider range of these different hypermedia types, including e.g. anchors, is unified to the term "HMObject". We also

[1] In this paper we use the terms "Groupware Component" or "component" to refer to mobile cooperative software tools. It is important to bear in mind the difference to other uses of the term "component", e.g. in [12] and [18].

use the term "HMObject" since we use the term "Component" for software tools. Figure 2 contains a part of a UML model of a classical hypermedia model (in light gray) and leaves out the associations in order to focus on sharing, semantic types and process support. The reader should note that the included part of a classical hypermedia model could of course cover a wider range of classes. Independent of the classical hypermedia model used, it is important for our approach that every hypermedia object is also a shared object with a unique identifier and shared attributes. In addition, shared objects can be made persistent. These provisions enable shared hypermedia workspaces [11] in which users access shared and persistent hypermedia objects, either synchronously or asynchronously, and interact with them through tools. Users interact with each other through state changes on shared objects (of course, additional informal communication channels can be used). By using suitable concurrency control and change propagation mechanisms this enables cooperation-aware applications since shared data is explicitly modeled.

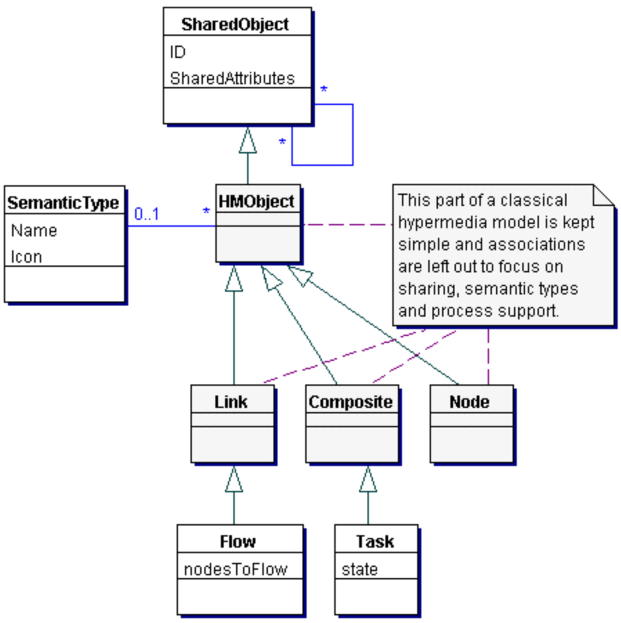

Figure 2: Shared Hypermedia with Process Support

In order to model processes, we incorporate task-related semantics such as state (e.g. *inactive*, *enabled*, *active*, *completed)* into a special composite class called *Task*, and control and data flow semantics into a special link class called *Flow*, such as which nodes contained in a source task composite should flow into a destination task composite after completion of the source task [29]. Hence, a process is represented as a set of hypermedia tasks connected by flows. Such a hypermedia-based process representation can contain associated materials because the hypermedia structure is not limited to tasks and flows. In this way, our approach combines process support with information management. Additionally, other node and link types can be incorporated, e.g. modeled users can be assigned to tasks or subtasks. Thus, this modeling provides an integral model of all relevant information. Also, the goal of a cooperation process can be explicitly represented, e.g., using textual annotations or links to the description of the goal.

Modeling the organization of the shared workspace as a task of the hypermedia model makes the organization of the workspace itself part of the shared workspace in the sense that it is shared information that can be worked on through synchronous or asynchronous cooperation (see R2). Therefore, there is a seamless integration of the organization task and all other tasks worked on within the shared workspace. Additionally, the way in which the work is organized can keep up with the evolving nature of shared workspaces because it can be tailored cooperatively and at run-time. Figure 1 shows three main parts of the shared object space. Each of the three parts uses a hypermedia structure with special node and link types. The content structure models the contents of a shared workspace, the different documents and their relations. The team structure models users and teams working in a shared workspace and the process structure models tasks and activities performed in a shared workspace. These structures are connected by links, e.g. a user should create a document or a task generates some documents.

Active Queries for Workflow Support. Workflow technology is intended to support work by enacting explicitly modeled and represented business processes [4]. Based on the process related semantics (incorporated in our hypermedia model) the modeled cooperative processes can be executed, e.g. setting a task's state to *completed* (by using the cooperative hypermedia editor) sets the state of following tasks to *enabled*. Active queries can be used for workflow support of the modeled processes. An active query provides up-to-date search results. It uses an observer mechanism [7] in order to notify the query model to execute the query again on changed objects of types queried for. If a user for instance would like to have a work-list presenting uncompleted tasks he or she is assigned to, an active query can be defined using a search tool and marking the query active. Hence, a query is not only based on our hypermedia model, but also on the process-related semantics. An active query enables the presentation, notification and the use of important information throughout the workflow. Queries can be reused, people can easily learn to create their own queries by participating in a query definition session or using a shared query model as a template and the workflow support is available synchronously.

Run-Time Extensibility of Hypermedia Object Types

When extending a set of given hypermedia object types in a shared workspace, two cases need to be considered. Firstly, a fixed set of hypermedia base types (e.g. node, link) can be extended. This may include extensions of the base types themselves (usually, this is not supported), or they may be categorized into semantic subtypes. Every hypermedia object can be assigned to a semantic type containing at least a name and an icon (see Figure 2). These types can be used to capture application or domain semantics. Only specific (semantic) types of links maybe allowed between specific (semantic) types of nodes. Secondly, while keeping the set of hypermedia base types stable new types of content (e.g. chunks of information managed by other applications) may be integrated via shared object references. These object references function as wrappers. They provide the functionality of shared hypermedia objects to the rest of the shared workspace while delegating display and manipulation of the content object to appropriate tools.

Run-Time Extensibility of Tools

When a team working on shared hypermedia objects in the shared workspace requires new tools, such tools need to be made available to all users in the team. In addition to *cooperative content editors* which are needed to edit the actual contents (shown as triangle, circle and rectangle in Figure 1) like a shared whiteboard, tools for facilitating orientation and awareness in the shared hypermedia workspace (see R4) and tools for explicit communication, e.g. chat or video conferencing tools, are needed (see R3). Over time, new and different tools might be required (see R5). Shutdown of all clients when adding a tool would not be acceptable to users in a session, who are working on a problem. Instead, if one user makes a new tool available, all users need to be able to use it immediately together. Here, we employ a component-based approach to cooperative hypermedia.

Groupware Components

Software components are physical packages of executable software with well-defined and published interfaces [25]. In [27] this has been applied to Groupware. In [27] a detailed discussion of the benefits of applying component-based technologies to the development of Groupware in general can be found. In this paper, we focus on benefits for cooperative hypermedia. Groupware Components are visually interactive tools, which allow users to access and manipulate shared data objects cooperatively or individually. Therefore, they work on shared data and are notified of changes regarding the shared data that they are interested in. These components are characterized by mobility, i.e. they can be loaded at run-time without forcing the user to shut down his or her shared workspace system as new cooperative tools become available. In addition to the tool itself, a tool package includes the shared data types, instances of which it works on. The whole package is uploaded to a common server so that it is automatically downloaded in case it is missing or outdated on a user's system. Therefore, new tools and data types (and the shared data in the shared workspace, of course) will be available for everybody without installation effort. The same is true for updates.

An instance of a Groupware Component works on one root shared object, which can point to several other shared objects (see Figure 3). Several other instances of Groupware Components can work on this root shared object (and also on any other shared object) at the same time, providing different functionality and views on the same shared objects. At any point in time, though, one instance of a Groupware Component is bound to exactly one instance of a shared object (the root shared object) by an instance of *ObjectComponentBinding*. A shared object in turn can be bound to several Groupware Components by several different *ObjectComponentBinding* instances.

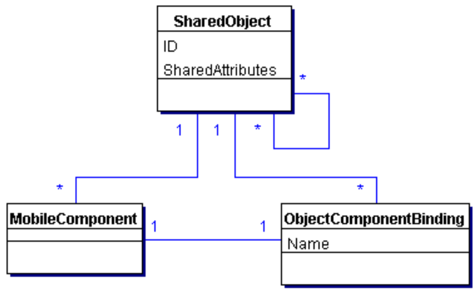

Figure 3: Groupware Components

Wrappers. A special first-class hypermedia type called *SharedObjectReference* can point to any shared object (see Figure 4). By opening such an object one of the associated instances of *ObjectComponentBinding* can be selected and the bound Groupware Component is invoked.

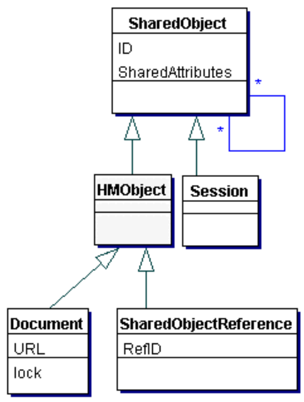

Figure 4: Wrappers

This model can also be used to represent and control cooperation state: Since sessions are also modeled as shared objects, an instance of *SharedObjectReference* can also point to a session. A referring session managing component can be invoked by opening this reference object, e.g. to join the session. Another special first-class hypermedia type called *Document* is used for pointing to non-shared documents by using a URL. This type provides locking and awareness possibilities for this kind of documents since concurrent access it not possible. But for supporting also synchronous work on these documents our approach proposes the integration of an application-sharing tool.

Tailorability. For the end-user, a shared object can show all names of associated instances of *ObjectComponentBinding* and dependent on which one is selected by the end-user the bound Groupware Component is invoked. The selection which component to invoke can also be made by the system dependent on information about the end-user, his or her infrastructure, activities or roles. With this tailorability support, the cooperative use of tools on a shared object is not limited to sharing between tools of the same type (e.g. all users use the same text editor to work on a shared text node). Instead, users can use different tools (i.e. components of different types) to work on a shared object. This makes perfect sense if users have different roles or preferences (e.g. one user might edit the composite node using a graphical browser component while the other use employs a textual view of all objects contained in the composite). Still, the users would share the same content, and the Groupware Components would display group awareness information to their respective users about the activities of the other user.

Tools for Facilitating Orientation. Our approach proposes three tools for facilitating orientation and awareness in the shared hypermedia workspace based on [11] (see R4). First of all, a cooperative hypermedia editor is needed for browsing and manipulating the hypermedia-based representation of the cooperative work processes. Content, team and process structures can be created and connected individually or cooperatively. Furthermore, the task as well as the control and data flow related semantics can be manipulated. This tool supports orientation and awareness in the cooperation processes because users can browse the task structure and get an overview about the state of work, e.g. which tasks have been finished, which are ongoing, etc. Additionally, this editor provides group awareness in the sense of providing information about who is working on which node (e.g. task, subtask, folder or document). Secondly, our approach uses a cooperative navigation tool for providing an overview about the entire hypermedia-based model. The cooperative hypermedia editor might only show the contents of a composite of the hypermedia structure and not the whole model. The navigation tool hierarchically presents the titles of all nodes and their composite relationships within the hypermedia model. It supports cooperative or individual browsing through the whole model. Regarding awareness the tool indicates who is working on which node by showing user names. Clicking on the node title in the navigation tool opens a suitable content editor on that node, e.g. for a composite node the cooperative hypermedia editor is opened. If others already use the node (see above), this user joins their cooperative working session. Finally, a cooperative search tool is needed for supporting queries on the shared workspace. This tool supports content related queries (e.g. which node(s) contain task XY?), team related queries (e.g. which tasks or documents is user A currently working on?) and process related queries (e.g. which tasks have already been completed?). This query tool allows cooperative definition and execution of queries in order to support teams working cooperatively on a task. For displaying the search results the navigation tool (see above) is reused.

The Prototype System

Our current prototype is called XCHIPS ("eXtensible Cooperative Hypermedia Integrated with Process Support"). It's implemented in pure Java using DyCE ("Dynamic Collaboration Environment"), a Client/Server-based Java framework for the development of Groupware Components [27]. Figure 5 shows the system architecture of XCHIPS. The DyCE server provides persistence for shared objects, shared object types and Groupware Components as well as management services, e.g. uploading and downloading of components, transactional support for accessing and modifying shared data. The client provides a desktop for accessing components and shared data. For XCHIPS, we have modeled shared hypermedia with process support as well as the presented wrappers on top of DyCE's shared object model. In addition, we have implemented several Groupware Components working on our model.

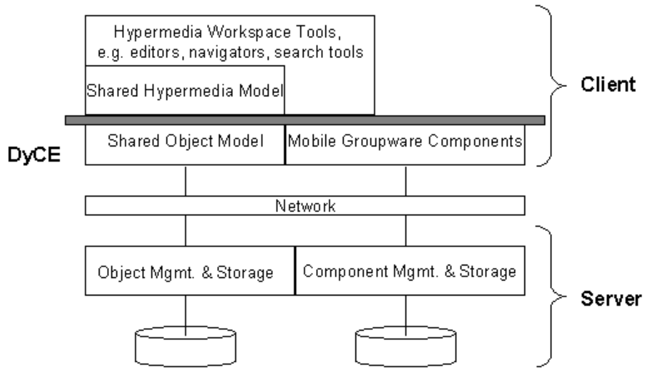

Figure 5: System Architecture

Our prototype system includes the presented cooperative tools for facilitating orientation and awareness. For supporting explicit communication (see R3) we have built a chat tool and an audio conferencing tool based on the Java Media Framework. Current cooperative content editors are a shared whiteboard, a shared Web browser, a shared text editor and a shared brainstorming tool.

Integration of an application sharing tool. Currently, we have integrated Microsoft's NetMeeting into our system in order to support application sharing on Windows platforms. This has been done by using a Java-COM bridge.

World Wide Web Integration. For Web integration (see R6) our current prototype client can run as signed Java applet. Therefore, it is assessable from Web browsers. Additionally, the hypermedia model can handle URLs. Furthermore, the server integrates a Web server so that any document can be uploaded to it. For this, an instance of *Document* provides functionality and automatically references the concerning document. In addition the web server provides an http-based API for invoking XCHIPS sessions.

A PROTOTYPICAL USE CASE

This section describes a usage scenario, which is prototypical for the usage of XCHIPS in the context of EXTERNAL. All usage scenarios in EXTERNAL involve cooperative work planning and cooperative (synchronous and asynchronous) work on shared objects, as demonstrated in the following description.

Alice and Bob want to write a progress report for a project they are working on. They work in a distributed setting and use XCHIPS as a shared workspace system. First of all, they use the cooperative hypermedia editor in order to formulate their shared tasks and processes explicitly. In parallel, they communicate about their work using the audio conferencing tool. They make some notes with the shared text editor. Secondly, they browse cooperatively into the task *write chapters* in order to assign themselves to the different subtasks. Figure 6 shows Alice and Bob working cooperatively with the hypermedia editor and composite navigator. Both components work on shared composite objects. The navigator currently provides a hierarchical overview about the nodes and composites of the root composite. The editor currently shows the contents of the task *Write chapters*: Tasks *Write chapter 1* up to *Write chapter 4* and representations of Bob and Alice assigned to the different tasks. Both components provide awareness information, e.g. Alice and Bob are working on *Write chapter 3*.

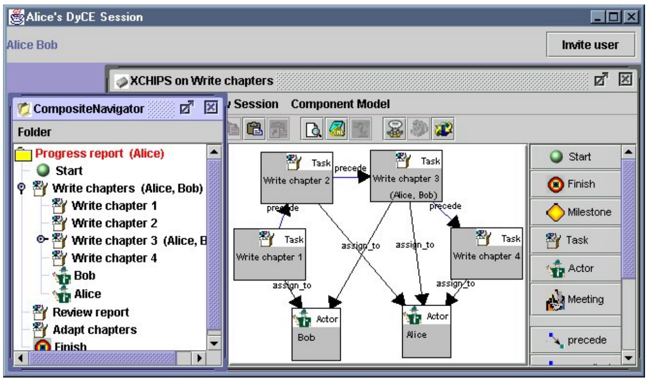

Figure 6: Cooperative Hypermedia Editor and Composite Navigator

The current object type schema of the cooperative hypermedia editor shows the Task composite with different semantic types, e.g. Start, Finish, Milestone, Task. These types point to the same object type, but are different semantic types so that all of these have a state, but different link types (e.g. *precede*) are allowed between them. Next, Alice and Bob use the cooperative text editor component for writing the chapters they are assigned to. They prefer using the cooperative text editor component in comparison with standard tools such as an office suite's word processor for instance, because this allows smooth switching between asynchronous and synchronous work to coordinate changes of the different chapters. They put the resulting shared documents into the linked tasks by opening the cooperative hypermedia editor on them and using a shared object reference pointing to the shared text editor object, respectively.

Regarding chapter 3 they are not sure about the topics. That's why they want to do some brainstorming. Bob has downloaded a brainstorming package and uploads it to the server. Immediately, it becomes available for both – Alice and Bob. Alice creates a shared object reference node in the *Write chapter 3* task for pointing to an instance of the brainstorming model (see Figure 7). By opening this node they get the referring component opened in the same session. Using the brainstorming component they collect topics for chapter 3. They define a milestone, which should be reached after completing this task. In addition, they put some figures in the Figures folder and make some notes with the shared text editor.

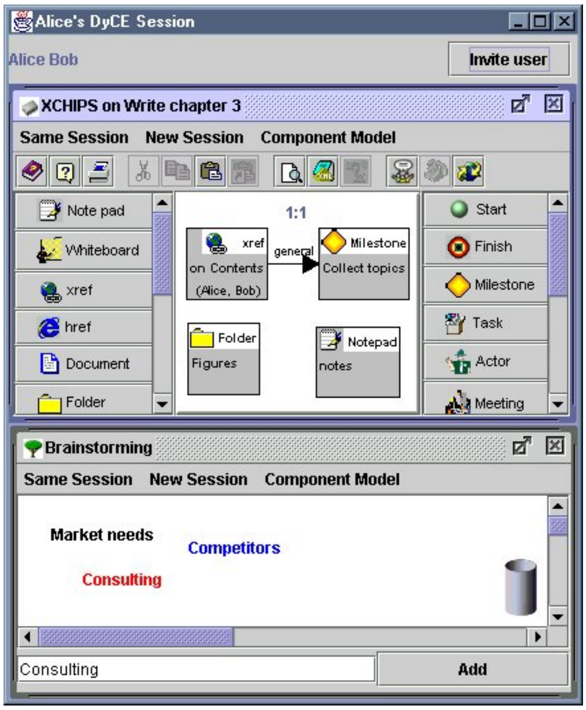

Figure 7: A Shared Object Reference to Brainstorming Content

After completion of each chapter they mark the belonging task *completed*. In order to have an always up-to-date overview about the tasks, on which they have to work, Bob and Alice create a work-list for themselves using the cooperative search tool and an active query. Figure 8 shows the definition of Alice's active query and the navigation tool showing its result.

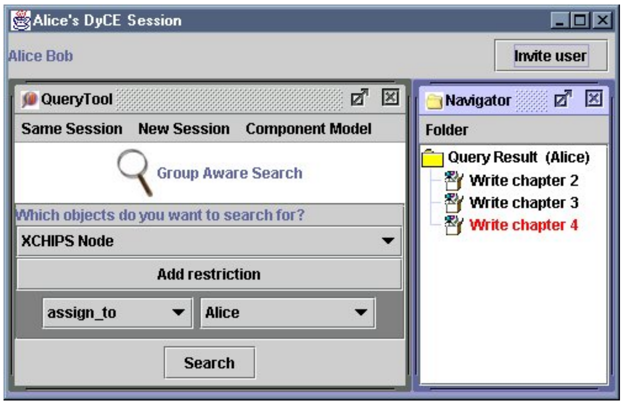

Figure 8: Cooperative Search Tool

Alice and Bob do this definition cooperatively so that Bob can now easily use Alice's query as template for his one. All he has to do is to clone Alice's query and to replace the value "Alice" by "Bob". After completion of all subtasks the *review report* task of the overall process becomes active. For this, Alice and Bob go through all chapters and read them cooperatively for simultaneous discussions. During their review task they make annotations in chapters where they want to change something. Additionally, they put references to these chapters into the *Adapt chapters* task by using shared object references. Finally, they work on the *Adapt chapters* task in order to adapt their report to their review. Again, they may create subtasks and assign these to team members.

COMPARISON TO RELATED WORK

There are in basically two areas of research that relate to work presented in this paper: Hypermedia and CSCW.

Hypermedia

In the hypermedia field, our approach is related to work done on the Dexter model, in the Open Hypermedia Protocol working group and many traditional and Web-based hypermedia systems. The Dexter model [12] focuses on hypertext structural constructs and their manipulations. It does not explicitly deal with the sharing and cooperative manipulation aspects of hypertexts. Our approach goes beyond the Dexter model as well as the core data model of the open hypermedia framework [20] by including shared hypermedia objects as well as bindings between shared hypermedia objects and Groupware Components. In the core data model of the open hypermedia framework a *collaboration service* provides two basic abstractions: *Session Record* and *Session State*. A session state is the only object, which might have shared attributes. The hypermedia objects themselves are not explicitly modeled as shared objects so that the sharing and awareness granularities are very low. Only special clients modeling their internal objects as shared objects and connecting to the Open Hypermedia Protocol have the possibility to provide fine-grained sharing and awareness support. In order to support more fine-grained sharing and workspace awareness, in our approach, in addition to sessions, every hypermedia object is also modeled as a shared object. As a consequence, all information to be shared can be modeled as shared hypermedia objects for direct manipulation or referenced by shared hypermedia objects for indirect access in a cooperative session.

There are several related systems, which support cooperation using the WWW. Purely Web-based systems use Web browsers as clients and Web servers with extensions, e.g. BSCW [2]. To support structuring of the information space BSCW only provides document lists in folders, but no task-related structure. Additionally, BSCW focuses on asynchronous work. Therefore, awareness mainly concentrates on document related events. Since version 3.2 of BSCW, synchronous awareness is provided through a user presence and activity monitor via additional Java applets. Communication is supported by a "talk" facility implemented using a Java applet. BSCW is not extensible regarding cooperative tools because these are not part of the shared workspace. Since BSCW is Web-based, every kind of document can be up- and downloaded and opened with an installed application using the helper applications specification of Web-browsers. However, these applications usually are single user applications so that synchronous work isn't possible. Additionally, on some machines corresponding applications might be missing and then users might not be able to access the documents.

The EuroCODE project [9] aims at a Dexter-compliant architecture for supporting sharing of hypertexts. The hypertext model provides many structuring possibilities, but processes are not explicitly modeled in the described system. It provides a variety of cooperation modes on shared documents, but focuses on asynchronous cooperation. It uses an event notification and locking mechanism based on an object-oriented database system. Synchronous work is supported by integrating the hypermedia architecture with an application sharing tool providing non-cooperation aware clients. For explicit communication a conferencing system is used. Software components for extensibility and tailorability of the cooperation environment are not used in this approach. Web-integration is not mentioned.

HyperDisco [31] supports shared workspaces containing hypermedia objects and external application files that can be accessed and manipulated by heterogeneous tools. It also provides an open protocol to integrate new tools into its infrastructure. It supports asynchronous cooperative authoring with heterogeneous tools. CAOS [21] is a component-based open hypermedia system, which focuses on spatial parsing and cooperation support. Posties is a WebDAV application for cooperative work [6]. It supports asynchronous cooperative authoring over the Web. Manufaktur [15] uses open hypermedia technology to integrate documents in a cooperative virtual environment with a 3D user interface. Open hypermedia systems, e.g., HyperDisco, focus on linking and integrating heterogeneous tools (and the information objects the tools work on) into an open hypermedia system. They also adopt a component-based approach for developing open hypermedia services (e.g., navigational hypertext service and spatial hypertext service), but this architecture focuses on interfaces and not on run-time extensibility. Compared to our approach, the open hypermedia protocol approach to tool integration is more light-weighted and open than using our Groupware Component model while our approach provides more tight and flexible tool integration needed by Groupware systems that support synchronous cooperation.

Cooperative Hypermedia systems like SEPIA [23] and CHIPS [29] provide flexible ways for structuring shared workspaces. Additionally, asynchronous as well as synchronous cooperative work is supported. Awareness is supported on hypermedia nodes and on some special issues of the whole workspace. There is no overview over the entire shared workspace structure. CHIPS supports semantic types, but it cannot be extended at run-time if the tailoring or extension of an object type or tool requires the modification of the system code. Both SEPIA and CHIPS are not Web-based; standard external documents cannot be handled.

CSCW

Leading commercial Groupware platforms for building shared workspaces like Lotus Notes [e.g. 26], Microsoft Exchange Server and Outlook [e.g. 3] provide only limited structuring features. Folders and views can be used for organization, but they show only a simple list of documents (a view shows documents fulfilling a specified query), and no task related structure. Moreover, both systems focus on asynchronous work. Therefore, workspace awareness support focuses on document related events. A possible integration of application sharing software does not support cooperation-aware clients. Regarding extensibility, it is not guaranteed that different applications are available in each situation (like our Groupware Components are). Finally, Web front ends to these systems provide only limited functionality.

Systems using the metaphor of rooms for grouping users and group-aware tools like TeamRooms [22] or CBE [14] are extensible: New tools can be added to rooms. However, for structuring the information space only tool and URL lists [14] are available. Persistent documents that are manipulated by tools can only be accessed via these tools and cannot be structured independently from them. In addition, the tools do not work on a shared data model, so that the tailorability is limited. Communication is supported via corresponding tools, such as a chat tool. Regarding awareness, user presence and telepointers [22] are provided. CBE is Web-based and therefore allows adding of URLs into rooms.

Semantically rich structures in the sense of process support are provided by classical workflow systems. However, classical workflow systems focus on asynchronous work. Shared documents are worked on using single user applications and process editors can't be used synchronously by multiple users. Up to now, synchronous cooperation with immediate interactive feedback has been added to classical workflow systems only for conference purposes like in [30], but not in order to work synchronously on shared objects. Web-based workflow systems often provide users only with a "ToDo" list and task descriptions, e.g. [1], or document lists with some document inspection functionality.

CONCLUSIONS AND FUTURE WORK

In this paper we have presented a component-based cooperative hypermedia approach, which addresses two problems that have so far not been addressed by related cooperative hypermedia as well as Dexter-based or OHP-based hypermedia systems:

1. Run-time extensibility of shared hypermedia object types, and

2. Run-time extensibility of cooperative tools operating on shared hypermedia objects.

By combining a shared component-based hypermedia workspace with an extensible set of object types and cooperative tools, and by providing specific hypermedia types for modeling and executing emerging workflow, all requirements of shared enterprise workspaces can be addressed. Current shared workspace systems have not provided adequate support to deal with the dynamic character of shared workspaces, i.e. the evolving cooperation processes, the evolving information structures, and the evolving set of cooperative tools needed by the users. In our approach, a shared enterprise workspace consists of a shared object space as well as a shared component space. We use shared hypermedia with process support for cooperation-aware applications and semantically rich information structures based on an integral model of all relevant information. Shared tasks are explicitly presented. Therefore, a seamless integration of the organization task and all other tasks within the shared workspace is provided. In addition we use Groupware Components and bindings between these and shared objects for providing easy extensibility and tailorability of the shared workspace system at run-time. Shared object references provide the glue between the organization structure and the actual content documents on which appropriate components can be opened. Workflow is supported by active queries and by automating data and control flow. Appropriate tools support orientation and awareness.

First usage experience with this approach in the EU project EXTERNAL indicates the applicability of our approach. However, a formal evaluation of the system in three user communities is currently performed. These real world use cases and their evaluations will provide feedback for improving the system. In addition, it is a challenge for us to integrate our approach with and contribute to standardized workflow services and open hypermedia.

ACKNOWLEDGMENTS

We would like to thank our EXTERNAL partners for inspirations and discussions as well as the anonymous reviewers for their very detailed comments.

REFERENCES

1. Ames, C.K., Burleigh, S.C., and Mitchell, S.J. A Web Based Enterprise Workflow System. In *Proc. of Group'97*, ACM Press, Phoenix, Arizona, 1997.

2. Bentley, R., Horstmann, T., Sikkel, K., and Trevor J. Supporting Collaborative Information Sharing with the World Wide Web: The BSCW Shared Workspace System. In *The World Wide Web Journal: Proc. of the WWW Conference*, Issue 1, O'Reilly, 1995, 63-74.

3. Byrne, R. Building Applications with Microsoft Outlook 2000 Technical Reference. MS Press, 1999.

4. Carlsen, S., Gjersvik, R. Organizational Metaphors as Lenses for Analyzing Workflow Technology. In *Proc. of Group'97*, ACM Press, Phoenix, Arizona, 1997.

5. EXTERNAL, EU Project, IST-1999-10091, New Methods of Work and Electronic Commerce, Dynamic Networked Organisations. Partners: DNV, GMD-IPSI, Zeus E.E.I.G., Computas, SINTEF, http://research.dnv.com/external/, 2000-2002.

6. Feise, J. Posties: A WebDAV Application for Collaborative Work. In *Proc. of HT'00*, ACM Press, San Antonio, Texas, U.S.A, 2000, 228-229

7. Gamma, E., Helm, R., Johnson, R., and Vlissides, J. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1994.

8. Goldman, S. L.; Nagel, R. N. and Preiss, K. (1995), Agile Competitors and Virtual Organisations, Van Nostrand Reinhold, 1995

9. Grønbæk, K., Hem, J.A., Madsen, O.L., Sloth, L. Systems: A Dexter-based Architecture. In *Communications of the ACM* 37, 2, 1994, 65-74

10. Gutwin, C., and Greenberg, S. Workspace Awareness for Groupware. In *Proc. of CHI'96*, Vancouver, 1996, 208-209.

11. Haake, J.M. Facilitating Orientation in Shared Hypermedia Workspaces. In *Proc. of Group'99*, ACM Press, Phoenix, Arizona, 1999.

12. Halasz, F., and Schwartz, M. The Dexter hypertext reference model. K. Grønbæk and R. Trigg, Eds. In *Communications of the ACM* 37, 2, 1994, 30-39

13. Keith Edwards, W., and Mynatt, E.D. Timewarp: Techniques for Autonomous Collaboration. In *Proc. of CHI'97*, ACM Press, Atlanta, GA, 1997

14. Lee, J.H., Prakash, A., Jaeger, T., and Wu, G. Supporting Multi-User, Multi-Applet Workspaces in CBE. In *Proc. of CSCW'96*, ACM Press, 1996.

15. Mogensen, P., and Grønbæk, K. Hypermedia in the Virtual Project Room - Toward Open 3D Spatial Hypermedia. In *Proc. of HT'00*, ACM Press, 2000.

16. Nielsen, J. Hypertext and Hypermedia. Academic Press, Inc., San Diego, 1990

17. Nomura, T., Hayashi, K., Hazama, T., and Gudmundson, S. Interlocus: Workspace Configuration Mechanisms for Activity Awareness. In *Proc. of CSCW'98*, ACM Press, Seattle, Washington, 1998.

18. Nürnberg, P.J., Leggett, J.J., and Wiil, U. K. An Agenda for Open Hypermedia Research. In *Proc. of HT'98*, ACM Press, Pittsburgh, PA, 1998, 198 - 206

19. Pallot, M. and Hof, C. Fast Reactive Extended Enterprise (FREE), European Society of Concurrent Engineering, conference on concurrent enterprising, 1997, 339-343.

20. Reich, S., Wiil, U.K., Nürnberg, P.J., Davis, H.C., Grønbæk, K., Anderson, K.M., Millard, D.E., and Haake, J.M. Addressing interoperability in open hypermedia: the design of the open hypermedia protocol. In *The New Review of Hypermedia and Multimedia*, Volume 5, Taylor Graham, 1999

21. Reinert, O., Bucka-Lassen, D., Pederson, C.A., Nürnberg, P.J. CAOS: A Collaborative and Open Spatial Structure Service Component with Incremental Spatial Parsing. In *Proc. of HT'99*, ACM Press, Southampton, U.K., 1999, 49-50

22. Roseman, M., and Greenberg, S. TeamRooms: Network Places for Collaboration. In *Proc. of CSCW'96*, ACM Press, Cambridge, MA, 1996.

23. Streitz, N., Haake, J., Hannemann, J., Lemke, A., Schuler, W., Schütt, H., and Thüring, M. SEPIA: A Cooperative Hypermedia Authoring Environment. In *Proc. of HT'92*, ACM Press, Milano, 1992, 11-22

24. Strømseng, K., Olsson, N., Haake, J., Scagno, G. Extended Enterprise Requirements, *The EXTERNAL project deliverable D1*, 2000, http://research.dnv.com/external/PW_Tools/PWD/3/31/D/3-31-D-2000-01-1

25. Szyperski, C. Component Software. ACM Press, Addison-Wesley, 1998

26. Tamura, R. Special Edition Using Lotus Notes and Domino 5. Macmillan Computer Publishing , 1999

27. Tietze, D.A. A Framework for Developing Component-based Co-operative Applications. Dissertation at TU Darmstadt, GMD Research Series, Germany, 2001

28. Wang, W., and Haake, J.M. Flexible Coordination with Cooperative Hypermedia. In: *Proc. of HT'98*, ACM Press, 1998, 245-255

29. Wang, W., and Haake, J.M. Tailoring Groupware: The Cooperative Hypermedia Approach. In *Computer Supported Cooperative Work: The Journal of Collaborative Computing*, Vol. 9, No. 1, 2000, Kluwer.

30. Weber, M., Partsch, G., Höck, S., Schneider, G., Scheller-Houy, A., and Schweitzer, J. Integrating Synchronous Multimedia Collaboration into Workflow Management. In *Proc. of Group'97*, ACM Press, 1997.

31. Wiil, U.K. and Leggett, J.J. Workspaces: The HyperDisco Approach to Internet Distribution. In *Proc. of HT'97*, ACM Press, Southampton, U.K., 1997, 13-23