

A Cooperative Visual Hypermedia Approach to Planning and Conducting Virtual Meetings

Weigang Wang¹, Jörg M. Haake², and Jessica Rubart^{1,3}

¹Fraunhofer Institute for Integrated Publication and Information Systems (IPSI)

Dolivostr. 15, 64293 Darmstadt, Germany
{wwang, rubart}@ipsi.fhg.de

²FernUniversitaet Hagen, Computer Science VI

Informatikzentrum, Universitaetsstr. 1
D-58084 Hagen, Germany
joerg.haake@fernuni-hagen.de
³go4teams GmbH

Julius-Reiber-Strasse 15a, 64293 Darmstadt, Germany

Abstract. Most distributed meeting support systems focus on meeting management and audio/video communication mechanisms. They provide little support for a flexible meeting process and a shared information space with structure-rich visual artifacts. In this work, a cooperative visual hypermedia system is developed to provide visual hypermedia artifacts for team members to manipulate in a distributed meeting. The visual hypermedia system is a hypermedia-based drawing system that integrates visual hypermedia artifacts and structures found in multiple hypertext domains. In addition, the visual hypermedia is integrated with process support for flexible meeting control and for easy setup of audio/video and application sharing communication channels. A use cases is presented, which shows that using the cooperative visual hypermedia, distributed teams can perform many kinds of meetings, in the meantime, enjoying dedicated support for the planning, control, information management, and follow-up activities of a distributed meeting.

Keywords. Cooperative hypermedia, visual hypermedia, process support, distributed meeting support.

1 Introduction

The trend towards globalization of business activities requires business teams to carry out their projects in a distributed environment. While face-to-face project meetings have always been central to effective teamwork, distributed meetings are used more and more often to reduce traveling costs and time, and to couple with emerging issues on short notice.

Traditional meeting support systems focus on augment co-located face-to-face meetings. In recent years, many distributed meeting support systems have been developed to provide audio/video communication channels, meeting management services, and some simple floor control mechanisms. However, these systems lack support for a well-planned flexible meeting process, lack a common information space with structure-rich visual artifacts, and lack the support for document handling.

Our approach to tackle the problems is to develop a cooperative visual hypermedia for providing

- a common information space with structure-rich visual artifacts,
- flexible meeting process modeling and control,
- integrated document management, and
- session-based meeting management and easy setup of communication and application sharing channels for a meeting session.

The remainder of the paper is organized as follows: section 2 provides a problem analysis on meetings, distributed meetings and distributed meeting support systems. Section 3 introduces the cooperative visual hypermedia concept and provides detailed descriptions of our approach to addressing the found problems. Section 4 describes hypermedia related implementation techniques that make our approach possible. Section 5 presents use cases of our system. Section 6 presents related work and section 7 concludes the paper with a summary, contribution highlights, and plans for further work.

2 Problem Analysis

Meetings are social or business events when a number of persons come together at a certain time and place for certain purposes, such as discussion and decision-making. Good meetings can create a dynamic interaction between participants to achieve more than one could achieve alone. Meetings are an important part of any business. There are many kinds of business meetings, such as,

- project work sessions,
- strategic planning,
- management team meetings,
- customer problem solving sessions,
- project reviews,
- staff meetings, and
- team building meetings.

2.1 What Makes a Meeting Successful

Working together does not always guarantee efficiency. Unnecessary or bad meetings could be just a costly interruption and a waste of time.

A good meeting needs to have [8]

- A strategically designed agenda with commonly understood goals and objectives.
- A clear, agreed upon process for reaching those goals and running the meeting.
- An awareness that people come with their personal preoccupations and feelings, as well as an interest in the subject at hand.
- A sense of involvement and empowerment;
- A skilled facilitator.

To make a meeting effective, five meeting specific roles need to be taken by meeting organizers and participants [11]:

- Chairman: who is responsible for meeting pre-coordination, setup, operation, and follow up; who decides the "why/what," "when," "where," and "who" of the meeting; and organizes and publishes a meeting agenda.
- Presenter: who finds out how much time the presentation is allowed to take, and provides information clearly and concisely.
- Recorder: who condenses information into key points, documents decisions, actions, and pertinent information, and produces meeting minutes ASAP after the meeting.
- Timekeeper: who helps maintain the pace of the meeting by reminding people of time used, and
- Participants: who shows up on time and prepared for discussion, and remains attentive, alert, and focused throughout the meeting.

The chairperson of a meeting needs to plan for support of each of these roles. It is possible to play more than two roles during a given meeting, but that is not wise in most meetings. Because it may not be possible for one to pay enough attention to the task at hand when one attempts to do this.

Therefore, to make a meeting successful, the following meeting support services should be provided:

- meeting planning for preparing a well-designed meeting process (or agenda) with a clear purpose in a business context. A meeting process or agenda specifies the sequence of meeting activities, the roles of participants, and the resources allocation to the activities.
- meeting management for setting up meeting facilities and for participants to take part in the right meeting sessions they are intend to,
- a common information space for information presentation and manipulation,
- leadership to run a meeting in an efficient and flexible way, and
- document handling for documents prepared before a meeting, presented and manipulated in a meeting, and stored for follow-up activities.

Based on their roles, meeting participants may have different access rights to the meeting support services. The common information space should not only provide tools for information presentation, but also allow meeting participants to annotate and modify the information presented (e.g. to modify an agenda). For supporting discussion and exploitation, widely used visual artifacts and tools should be provided.

2.2 Distributed Meetings (Virtual Meetings)

Distributed meetings are Virtual Meetings that denote the situation where meeting participants from different locations and organizational units meet in a "virtual" meeting place. In such a distributed virtual setting, meeting planning, coordination and communication become even more crucial. The meeting process as one fundamental component of collaboration needs to be viewed and supported as a part of the overall business processes taking place in the organizational and social context. Audio/video channels have to be easy to setup and use. It should also be easy for meeting participants to find the virtual place and join their part of the meeting, or be on-line at the right time for meeting organizers to bring them into the right meeting sessions. Before or during the meeting, the participants may have a need to import pre-existing documents into a common workspace, jointly view and manipulate the workspace in the meeting, and export documents to a shared organizational repository. As a discussion aid, synchronous manipulation and structuring of visual artifacts including writings and drawings are also very important.

2.3 Distributed Meeting Support Systems

Efficient and effective technology support for distributed meetings is a key to virtual organization effectiveness through

- (1) saving of cost and time required for communication and collaboration in distributed teams. This will enable the user organization to better cope with dynamics of the market.
- (2) better reaction on short notice will be possible. This leads to greater flexibility.
- (3) establishing/strengthening business relationships (not only to partners, but also to suppliers and customers) due to better communication and collaboration facilities.
- (4) reduction of media breaks due to integrated document management, collaborative work and conferencing facilities.

However, so far, attempts of technological support for distributed meetings (e.g., video conference studios for meetings, Internet-based and Web-based text or audio chatting, white boarding, application sharing tools) have been disappointing. Insufficient progress – or even failure – of most current meeting support systems are caused by their lack of support for shared document handling, by missing support for a rich set of information structures and visual artifacts, and by their insufficient integration with asynchronous modes of co-operation and related applications (e.g., workflow management systems). Most importantly, the current generation of tools for distributed meetings lacks a comprehensive model of such meetings in a business context. A model that integrates the meetings into the overall business processes is necessary to guide the construction of effective virtual meeting facilities. Because of all these deficits, the current generation of facilities remains of marginal value to businesses.

3 The Cooperative Visual Hypermedia Approach

Our approach to tackle the above mentioned problems is to develop a cooperative visual hypermedia for providing

- a common information space with structure-rich visual artifacts,
- flexible meeting process modeling and control,
- integrated document management, and
- session-based meeting management and easy setup of communication and application sharing channels for a meeting session.

In this work, we extended our component-based cooperative hypermedia system XCHIPS [7] with visual hypertext artifacts and above mentioned meeting support services. The key points of our approach are presented in the following subsections.

3.1 Visual Hypermedia: A Hypermedia-Based Drawing System with Structure-Rich Visual Artifacts

The visual artifacts and tools that people widely use in face-to-face meetings include blank sheets or whiteboards, on which people can write and draw using colored lines, shapes, and sticky tags or pin cards. Sticky tags or pin cards allow people to post them in one way and then to organize or classify them in different ways. It is also very common to see people to draw arrow lines between cards or group of cards to indicate their relationship. Cooperative electronic whiteboards are tools provided in many distributed meeting support systems for such purpose. But these whiteboarding tools lack the flexible structuring support, such as the composing and decomposing capabilities that many object-oriented drawing systems have and the reorganizing and classification, nesting, and linking that supported by many spatial [12] and graphical navigational [17] hypermedia systems. Spatial hypertext uses spatial layout and some visual characteristics to represent relationship. For instances, similar information objects are placed near to each other. Navigational hypertext is the classical hypertext that is normally presented to users as a content page with highlighted embedded links (e.g. a Web page). Navigational hypertext structures may also be presented visually as hypertext maps, in which labeled square or icons represented nodes and arrow lines between them represent links.

Our approach to providing structure-rich visual artifacts is to combine the features found in spatial and graphical navigational hypermedia systems as well as those found in drawing tools into a visual hypermedia system. Such a system can make use of various visual properties found in drawing systems for the expressing relationship among various visual artifacts. These visual properties include the (foreground and background) colors, patterns, images, the label text, the size, the orientation, the position, the shape, and the Z-ordering of the visual components. In the mean time, it makes all these visual components into hypermedia objects that can be re-organized into different sets or nested structures, be linked to or referred to each other, and linked to external documents.

For instance, while explicit link representations (e.g. as arrow lines) are good for expressing the precedence relationship between tasks, color-coding is better for indicating the state of the tasks (e.g. yellow means ready and green means on-going). Such visual hypermedia can be used to create and manipulate both the meeting process structures (see section 3.3 and section 5) and the visual contents (see section 5) that used in various sessions of the meeting .

The user interface (i.e. the XCHIPS browser) for the visual hypermedia (see Figure 1) resembles that of a drawing tool: it has two visual object palettes providing pre-defined visual hypermedia object types and general types that can be used to derive new types. In Figure 1, on the top toolbar, there are buttons to activate a textual chatting tool and the MS NetMeeting tool. On the right-hand side palette, there is a list of visual object types for process modeling. On the left-hand side palette, there are object types for creating information objects, among which there is a general node type (the button with a blue “i” icon) and a general link object type (the button with a link icon) for users to create new node and link types on selected hypermedia model objects (such as “folder” or “href”, see section 4.1 and 4.2). The appearance of new visual types can be defined by using menu operations on the newly created general types, for instance, to change its foreground and background colors or patterns, the label text, the size, the orientation, the shape, and the Z-ordering of the visual component, or to load an image. These visual properties correspond to the visualization design dimensions that a good visual design should consider [16]. Any visual objects in the content pane can be used as a prototype for creating new instances of the same type with the same appearance (by copying). A view copy mechanism can be used to create multiple views of the same model object (see section 4.1). The whole nested structures rooted from a composite node can be used as a template for cloning the whole hypermedia structure in one operation.

3.2 Session-Based Meeting Management and A/V Setup

A meeting is normally organized into one or more sessions. A session is also a meeting, which may involve all or part of meeting participants. Sessions may happen in parallel or one after another. Each session takes a period of time used for a particular purpose or activity and happens normally in one meeting room. To simplify meeting management for distributed meetings, we do not make a difference between a meeting and a session. In our approach, a session is made up of a collection of people, groupware components, and shared objects. To provide a sense of virtual place, each session is visualized as a separated session window. Under the menu bar of the window lists all the names (uids) of the on-line participants of the meeting (see Figure 2). Groupware tools used in the session will be opened in the session window. All participants of a session will have a synchronous instance of the session window. Working with the XCHIPS browser component in a session, other groupware components invoked from the browser user interface or from the shared objects in its content pane can be activated using menu operations on the (closed) views of the shared objects. Options of

the menu operations are provided for users to open them in either in the same session window (thus becoming accessible to all users within this session) or in a new session window (providing a simple transition to individual work or a subgroup meeting in a new session). If a meeting takes a long time and needs multiple sessions, we use a meeting process structure to organize it (see next section) and to divide sessions by opening composite objects for different session activities in different session windows. Meeting organizers can invite other users to join their working session by activating the *Invite user* button at the upper right corner of the session window (see Figure 2). Users can also use a query tool to search for all the active meeting sessions and request to join selected sessions.

With the XCHIPS browser, it is very easy to set up a NetMeeting session for audio/video communication and application sharing for all the on-line participants (see the section 4.4 for detail). Users can also activate a Chat groupware component for all the session participants by clicking on the Chat tool button on the user interface.

3.3 Process-Based Meeting Control and Document Handling

A process structure can be described as a directed graph consisting of a coordination structure among the steps. Such a graph can be modeled as a typed hypermedia structure (see also [23]). In XCHIPS, the process structure is represented by a set of pre-defined hypermedia object types:

- Task nodes, which may carry a description of the task and other information objects (such as embedded links to external documents) as its contents, and which can be related to other resources via specific explicit links, such as a “assign” link from a role (or position). When a task node contains other task nodes connected with process links, it becomes a composite task node that presents a process or a sub process. Thus, processes can be nested and may form a layered graph. This nesting allows for representing the decomposition of work procedures into smaller parts. Task nodes have process-specific attributes, such as state and time, and have process-specific behaviors, such as state transitions and activation of certain actions,
- Process links, which represent precedence relationship between task nodes. Process links have control flow and data flow behaviors. For example, instances of the process link type “precede” can be used to link tasks and processes into sequences or parallel structures, where each precede link specifies under which condition the successor node can be activated and which documents will then flow from the predecessor node to the successor node,
- Role (or position) nodes, which specify the responsibility in a process or an organization,
- Person nodes, placeholders that present actual individuals,
- Assignment links, which assign a role to a task.
- Filled-by links, which specify which Roles are assumed by which Persons, and

- Any other nodes or links for additional information, such as information about the goals and objectives of the work process, background information, and resources.

Thus, in such a typed hypermedia environment, a process can be represented as graphs consisting of typed hypermedia objects, together with additional information related to the process.

For a clear overview of the state of a process, the state information of task nodes is color-coded in their views (white for inactive, yellow for enabled, green for active and brown for completed). Tasks can be performed by menu operations activated on the task nodes. Selecting "enable" and then "activate" menu operations on the root composite task node of a process structure can start the process execution. When a composite task node is "activated", the starting tasks of the process structure nested in the composite task node are "enabled". They can then "activate" their tasks. When the work of a task is finished, its actor can select the "complete" menu operation on the task node, such that its following tasks (or nested starting tasks if it is a composite task node) are "enabled". A process is completed when all its subtasks are completed.

Using such a process-enable hypermedia structure, a meeting process can be created:

- A meeting process can be modeled using a root task node, which contains other task nodes for meeting sessions and agenda items (meeting activities).
- Information objects (see section 4.1-4.3) presented and manipulated for each meeting activities can be placed into the task nodes represent the meeting activities before the meeting takes place.
- Meeting specific roles, such as Chairman, Recorder, Presenter, Timekeeper, and Participant can be modeled using Role nodes.
- Meeting participants can be modeled using the Person nodes.
- The meeting roles can be assigned to meeting activities by the assignment relationship.
- The meeting role and participant relationship can be specified using filled-by links.

The document handling capabilities are integrated into several types of embedded links that can be placed in folders or tasks in a process model (see section 4.12-4.3). The meeting process structure can be modified at the beginning of a meeting or modified on the fly during the meeting. The execution of a meeting process (i.e. the transition from one meeting activity to another) can be manually controlled by the Chairman. Presenter can open documents in his or her task node when the task is enabled. For annotation, or discussion and exploitation activities (tasks), visual artifacts can be used together with the audio or textual chatting communication between meeting participants.

4 Implementation Details

The cooperative visual hypermedia as described in previous section has been implemented and supported with a cooperative visual hypermedia browser in our component-based XCHIPS system, which is an extension of the CHIPS system [22-24]. This section focuses on the cooperative visual hypermedia related implementation issues.

4.1 Visualization and Behaviors of Hypermedia Objects

A graphical view of a hypertext is a view of its underlying model (instance). It is composed of the views of the objects (nodes) and relationships (links, containment, and visual indications) in the model.

If a node has content or if it is a reference to another object, the node will have at least two views: one for its closed view and the other for its opened view (for its content or the referred object). A node that refers to another object or external document (URL) can be seen as an embedded link. In this case, node opening is equivalent to (reference) link following.

Each node or link object may have more than one view. In this case, all of them are kept consistently up-to-date with the model of the node or the link. The view and controller object for a closed view has an instance variable pointing to its model object, while the model object has no direct (instance variable) reference to its view object. The changes to a model object are propagated to its view objects by means of an event subscription mechanism.

The controlling operations to the hypermedia object can be triggered using the menu available from the view objects. Menu operations available from a link view or a closed node view object include operations collected from both the view object and its underlying model object. We use Groupware Component to implement the opened view of a node. The binding between the model object and the view object for its opened view is resolved at run-time through a model-task-component triple object that is published on the view component object class (see [7] for details).

In addition to the explicit links, the layout (e.g. positions and visual properties) of the nodes can also indicate relationships. For instance, for a composite node presenting the structure of an organization, the contained views of nodes and links can be laid out as an organization chart – a top down hierarchical structure whose links are lines with rectangular corners. The content view of a composite node can be provided with a layout manager to automatically layout its components. For manually handled *free layout*, the positions, bounds and other visual properties are persistently stored so as to assure a stable spatial and visual relationship representation.

4.2 Text, Hand Writing or Drawings, Lines, Shapes, and Images as Hypermedia Objects

By default, these view objects are created together with a simple model object, which has a type label and a name attributes. The model object can be changed to other pre-defined model object types. The pre-defined types include composite model objects types (such as “folder” and “task”) which can contain other node and link objects, an xref model object that can refer to another shared object (see [7]), a href model object that can refer to a read-only external document (see below), or a Webdec model object which can refer to an editable external document (see below).

4.3 Shared Objects as Embedded Links to Shared Documents

For integrating document handling into the common information space, two types of shared objects are developed as embedded (cooperative) links to documents on the Web.

- href is an embedded link for a read-only access to documents on the Web (such as HTML pages and other document types that can be handled by plug-ins of a standard Web browser). The pages or documents are fetched and displayed by the standard Web browser or its plus-in applications. This embedded link is presented as a labeled icon. The behavior of the link is encapsulated in the href object type. Users can activate a menu upon the labeled icon to
 - set a label,
 - set a URL,
 - open a page or document for oneself, or
 - open it for all the active users in the meeting session.

Such a link that opens an external document (or a page) for more than one person (on a number of dynamically selected machines) can be considered as a kind of computed cooperative link. Both the 'open for one' and the 'open for all' methods can also be considered as process links (see the HB/SP series [29] on links and anchors as processes), as they both activate a method (or methods) of a COM object on a local machine (through a Java-COM bridge). The difference between them is that the 'open for all' method sets the 'showForAll' attribute of the href object; while the 'open for one' method not. After the invocation of the 'open for all' method, the changed attribute 'slot' is replicated to all the local proxies of the href object for its distributed users. This value change (i.e. from false to true) in turn triggers a local call to COM APIs. In this case, it is an href method calling for the FollowHyperlink(url) and close() methods of a Word.Document COM Object.

The open for all option can be used to do a Web-based presentation using a standard Web browser. For doing so, user can create a sequence of href embedded links pointing to each of the Web pages, and then activate them in a sequel in his or her presentation.

- Webdec is an embedded link for a read-write access to documents on the Web (such as HTML pages and other document types). The pages or documents are fetched and displayed by the Web browser for HTML pages and MS Office tools for MS office documents. This embedded link is presented as a labeled icon view. As the underlying data object of the view is a shared object, many attributes can be defined for the object and be reflected on the view for group awareness purpose. The behavior of the link is encapsulated in the webdoc object type. Among many operations on the object, users can activate a menu upon the labeled icon to
 - set a label,
 - set a URL,
 - check-out and edit (i.e. lock, download, and open) a document from an organizational shared document repository, which can be any Web server that supports the http get and put protocols, and
 - check-in (i.e. upload and unlock if locked) a document to the shared repository.

When a document is checked out for editing, it is locked and a name label appears on the graphical view of the embedded link. The name label provides a kind of group awareness information on who is editing the document. If another person tries to invoke any of the above menu operations, he or she will get a warning message. This message tells him or her that the document is locked and he or she can contact the person for releasing the lock or for setting up a NetMeeting session to share the MS Office application for the document (see next section). The locking is implemented with a locking-service that is a simple implementation derived from the naming service of the XCHIPS system. The upload and download methods used by the check-in and check-out menu operations are implemented initially using the common http get and put methods, and then in a later version using a WebDav [3] based Java package from one of our project partners. The open document method used in the Check-out menu operation is implemented using a COM API, i.e. an open() method of a Word.Document COM object.

4.4 Mobile Groupware Components as Embedded Links to Distributed Applications

We use Microsoft NetMeeting for audio/video communication channels and for application sharing (e.g. to share a PowerPoint presentation or to jointly edit a Word document). Because NetMeeting uses a point-to-point network connection, to setup a NetMeeting session for a group of people, the meeting host has to inform all the participants the host machine's IP address and let them call this address using a NetMeeting client. This is a cumbersome manual process. To make this simple, we develop another type of embedded link, with which the NetMeeting connection for a group of people can be setup easily.

Again, here we used a shared object as a wrapper of the NetMeeting COM APIs. Because in this case, we want to allow all the meeting participants to initiate, join or

leave a NetMeeting session, we need to provide them with a user interface to invoke the corresponding COM APIs. For this reason, the shared object has two views: a closed view (shown as an icon) and an opened view, which presented in a separate window (i.e. an Internal Frame within the session window). The closed view is used to invoke the opened view, while the opened view in the separated window provides the graphical user interface for users to setup and control a NetMeeting session.

The groupware component can be considered as a computed multi-source process link to the NetMeeting COM objects on the distributed machines of the meeting participants. When one of the participants activates the embedded link and then push the 'host a meeting' button on the opened view, the user's machine becomes the NetMeeting host (i.e. the destination of the NetMeeting link). Then he or she can push the 'start a meeting' button to let the NetMeeting client on each all participants' machine to connect to host automatically (see Figure 2).

In this connection process, the link sources (i.e. the NetMeeting clients represented as the IP addresses of all the participants are found from the active user list of the session object that is accessible from the shared object. This is possible as the User object of the system has an IP attribute whose value is automatically assigned when the user is logged on the system.

The NetMeeting connection call (through an NetMeeting COM application programming interface with the host IP address as its parameter) is invoked on the local instance of the groupware component (see [7] for detail). When each client invocation is done from each participant's machine, the NetMeeting host participant gets a dialogue box asking him or her to accept the NetMeeting connection from the client. After that the original NetMeeting control window appears on all the participant's desktop and the NetMeeting setup is done. Participants can talk to each other and they can also use this control panel to initiate application sharing, passing control to shared application.

Using the same techniques, an embedded link (and a groupware component) can be created for doing a PowerPoint presentation for the meeting participants. In this case, the embedded link is a multi-target link; its groupware control panel will has buttons to navigate forth and back in a slide show. This can be used together with conference phone or chat tools in situations when a NetMeeting connection is prevented by fire-walls at participants' sites.

5 Sample Use Cases and User Experience

EXTERNAL is an IST project on technology support for virtual enterprises. There are five partner organizations distributed in Germany, Norway, and Greece. XCHIPS is one of the tools developed in the project for providing 3C (coordination, cooperation, and communication) and cooperative work management services. The support for distributed meeting is one of its most important services. For testing our own ap-

proach, the XCHIPS system along with other tools developed in the project have been deployed and used in three use cases, one of which is for the joint planning and management of the project. After a period of testing, the first deployment of the XCHIPS system for use cases was in last June, and then a series of revised versions were issued based on the feedback from its use.

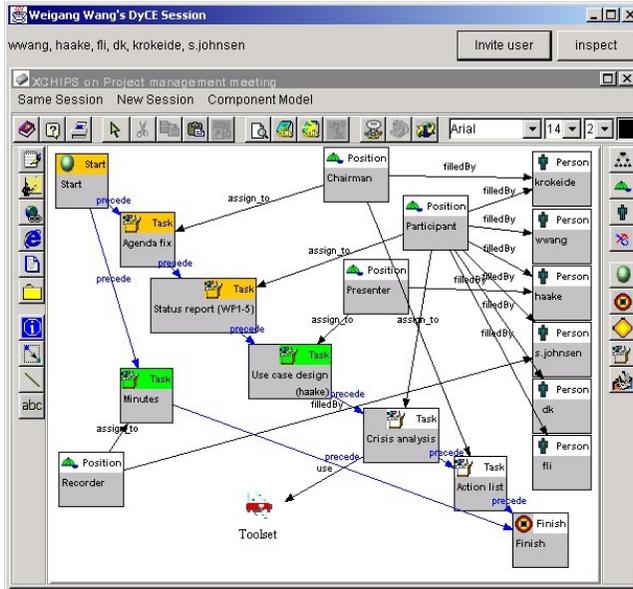


Fig. 1. A meeting process description

The following is a description of a distributed meeting use case using the XCHIPS system by the EXTERNAL project management team for distributed project management meetings. The project management team consists of the project coordinator, the work package managers and the site managers of the project. The following use case describes one of the project management meetings.

The purpose of the management meeting is to review progress, discuss use case designs, and perform a crisis analysis for identifying important issues that have to be addressed for a successful project. Before the meeting, the organizer (Chairman) of the meeting creates a meeting agenda (in the form of a meeting process structure in a composite task node, which is part of the overall process model for the joint planning and management of the project). The meeting process model includes the status report, user case design, and crisis analysis as its major activities (see Figure 1). Then the Chairman sends an email message to all the participants to inform them the time, the topics, and the (virtual) location of the meeting (i.e. the URL of the meeting process model). Based on the plan, the Presenter of the meeting prepares and then checks-in a PowerPoint file into the “User case design” activity that is assigned to him.

When the meeting time comes, the participants are all logged on to the system, and then are invited into the meeting session by the Chairman. Then one of the meeting participants sets up the NetMeeting tool for all the participants (see Figure 2 for the groupware control panel). As the first meeting activity, the Chairman tries to see if there is any request to adjust the meeting agenda. Then each work package manager reports the status of his or her work package using the audio communication channel.



Fig. 2. A groupware component for NetMeeting

After that the Presenter activates his PowerPoint presentation by selecting the "check-out" menu operation upon a Webdoc embedded link in side of the "Use case design" task. The Presenter shares it with others using the NetMeeting control panel on his desktop. After the presentation and a brief discussion, the Chairman starts the "Crisis analysis" task and opens the task node in the same session for all the meeting participants. Coordinated by the informal audio channel, each participant first posts the issues he or she thinks to be important (see Figure 3), and then they move the vis-

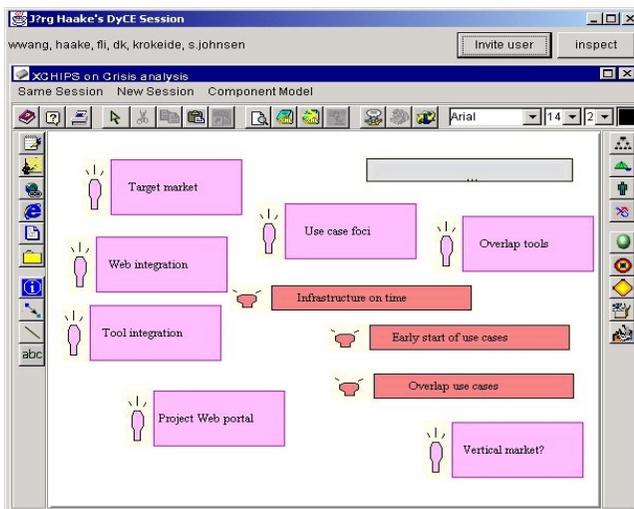


Fig. 3. Collecting important issues

ual cards into several clusters and using explicit links to indicate the relationship among the clusters (see Figure 4).

Finally, together with all the participants, the Chairman makes an action list to be carried out after the meeting. During the whole meeting process, the Recorder makes minutes individually using text, scribbles, or other visual components in the content pane of each activity. After the meeting, the Recorder collects his or her notes into a minute (accessible in the Minutes task node) and informs all the participants to see if there are any questions on it. Based on the decisions and action list made in the meeting, the project manager can monitor the execution of the follow-up actions.

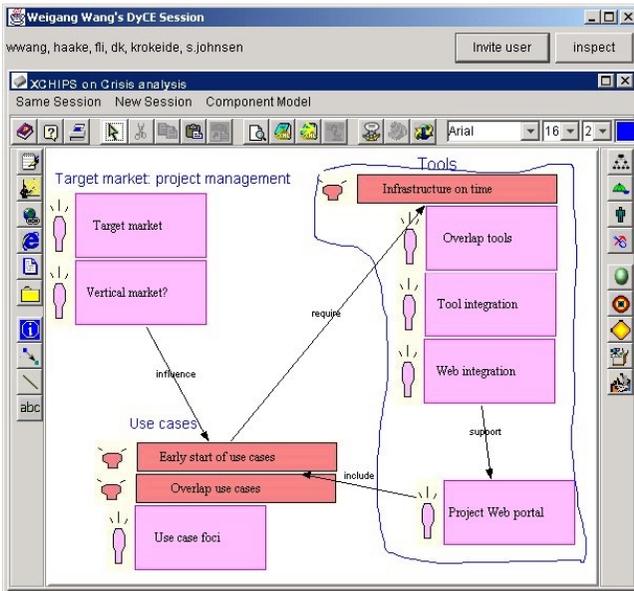


Fig. 4. Clustering and linking ideas

From over a year testing and half a year use of the system in the EXTERNAL project, we got numerous comments and feedback from the users. They liked the color-coding of the meeting agenda items (i.e. the task nodes) and the minute recorded within or accessible from the task node for each meeting activity. As this allows them to see (particularly for the late comers of a meeting) what has been performed, what is on going, and what are the next to come. Other team members who are not able to attend the meeting can navigate the meeting process structure to learn what happened in the meeting and to find and understand the decisions made in its original meeting context. They also liked the ability to write, scribble, post-visual artifacts, and to reorganize them for on-line discussions. The handling of MS documents and its integration into a process structure were considered a useful and practical solution, as all the user organizations of the use case partners of the project have MS Office suite and Windows operating system used in their daily working environment.

Users have also experienced many problems on using the system. As both the XCHIPS and the NetMeeting require the user organizations to open certain ports for inbound TCP/IP communications. One of the problems is with the firewalls. We resorted to the help of the system administrators and even the management of the organizations for opening the port or to deploy machines in a de-military zone of their Intranet settings. But still we have got one user organization refused to do so. For these reasons, in many situations, we have to use conference phone instead of NetMeeting for voice communication. This also motivates us to develop Web-based presentation tools (as mentioned in section 4.3-4.4) that do not have a firewall problem. Another experience we got is that even a NetMeeting session can be setup, participants still prefer to use conference phone for voice communication, as it provides a better sound quality. In these cases, the common workspace provided by XCHIPS was considered by most meeting participants a good aid for presentation and discussion. Our experience suggests that the system is better suited for short distributed meetings of a small group of people. We still have a long way to go before we can use such systems to replace long lasting (i.e. over one day) face-to-face project meetings of a large group of people.

6 Related Work

The hypertext technology most closely related to our cooperative visual hypermedia is spatial hypertext, such as the spatial and visual hypertexts supported by Aquanet, Viki [12], VKB [5] and CAOS [18]. Our cooperative visual hypermedia supports a kind of mixture of multi-domain visual hypertext structures, including spatial, set-based, navigational, automaton-based, and cooperative hypertexts. This kind of hypertext extends spatial hypertext with a set of ready-made visual hypermedia objects that is not only visual, but also has pre-defined computational semantics (i.e. each view object has an application specific model object behind it). It also provides ad hoc visual hypermedia objects (text, hand drawings, filled or unfilled shapes, images) created-on-the-fly in a shared information space. It supports multiple views (i.e. view objects) on the same model objects. These view objects can form multiple classifications or groupings (i.e. alternative views) on the same set of hypermedia objects. In addition, it includes explicit links that are missing in set-based spatial hypertext. These explicit links can enrich the explicit relationship expressiveness and the computational semantics (as demonstrated by the process links) of spatial hypertext. Other enhancement to existing spatial hypertext is the cooperative manipulation of visual artifacts and the cooperative document handling upon Web-based repositories.

Among meeting support systems, the one nearest to ours is the Dolphin system from our group a few years back [14]. Dolphin addressed a much wider range of meeting support issues, such as user interaction on large display in a meeting room and support meetings held in multiple meeting rooms. Dolphin also provides a hypermedia-based whiteboard as a common information space. While XCHIPS focuses on desktop based virtual meeting support for distributed teams. What XCHIPS adds to

Dolphin and other cooperative hypermedia systems developed in our group ([22-23, 7]) are the integrated process support, document handling, meeting management, and a rich set of visual artifacts for supporting distributed meetings.

Nest, in the sequel, we discuss several areas of meeting support systems in relation to our approach and our XCHIPS system:

- Desktop Teleconferencing: In addition to available desktop-teleconferencing functionality as provided by NetMeeting and WebEx, our approach also provides meeting oriented conference control as well as meeting specific functionality (e.g., agenda or meeting process planning).
- Video-conferencing rooms (e.g., media spaces [10]): XCHIPS may also be used in electronic meeting rooms with large displays and at the same time it can integrate well remote desktops for participants not available in the rooms.
- Shared documents (e.g., asynchronous shared workspace systems BSCW [2] and WebDav [3]; cooperative hypermedia systems KMS, rIBIS; meeting room systems Group Systems [15], CoLab [20]; Shared editing tools of GROVE [4], ShrEdit [13]; and shared drawing tools of GroupSketch [6] and CaveDraw [9]): Whereas shared document systems enable shared viewing or shared editing of isolated documents, XCHIPS exploits CSCW and hypermedia technology to facilitate rich information structures and seamless process support through the provision of shared workspaces, including traditional linear documents and hypermedia structures as well as support of transitions between different meeting phases. Furthermore, the use of multimedia functionality inherent in hypermedia will support the integration of different pieces of information without introducing media breaks.
- Asynchronous collaboration support (e.g., bulletin boards, e-mail or workflow management systems): In contrast to asynchronous collaboration tools, which can only support the exchange of information or describe schematic, well-structured processes, XCHIPS bridges the technology gap and integrates schematic workflow with unstructured phases of the business process, where flexible ways of problem solving, coordination and collaboration support are required.

7 Conclusions

Business processes of today are characterized by a change from individual work to teamwork as well as by a shift from single, independently acting corporations to virtual organizations operating as dynamic networks. To realize the benefits of globalization and business networking, we have to find a way for teams of knowledge workers effectively work together across distances and time zones. Efficient and effective technology support for distributed meetings is a key to virtual organization effectiveness. However, current distributed meeting support systems lack support for a well-planned flexible meeting process, lack a common information space with structure-rich visual artifacts, and lack the support for document handling.

Our approach to tackle the problems is to develop a cooperative visual hypermedia for providing

- a common information space with structure-rich visual artifacts,
- flexible meeting process modeling and control,
- integrated document management, and
- cooperative session-based meeting management and easy setup of communication and application sharing channels for a meeting session.

Two use cases are presented, which show that using the cooperative visual hypermedia, distributed teams can perform many kinds of meetings, in the meantime, enjoying dedicated support for the planning, control, information management, and follow-up activities of a distributed meeting.

In terms of hypertext technology and its application to distributed meeting support, our approach has a number of innovation aspects:

- The cooperative visual hypermedia integrates many visual and interaction features found in multiple hypertext domains including cooperative hypermedia, spatial hypermedia, navigational hypermedia, and automaton-based workflow hypermedia [21–23].
- It demonstrates the possibility and usefulness of the co-existence of multi-domain visual hypermedia structures for supporting distributed meetings.
- It provides flexible meeting process support and a common information space for meeting planning, flexible meeting process modification and execution, and follow-up activities (to be carried out in either synchronous or asynchronous cooperation). I.e. it provides a seamless transition between synchronous cooperation and asynchronous cooperation.
- It integrates document handling into the meeting process structures. More importantly, it can connect and integrate multiple meeting processes into the whole organizational business (process) context.
- The cooperative embedded links for opening external documents for multiple distributed users and the cooperative embedded links for controlling external applications cooperatively are new to existing hypertext linking mechanisms.

To be successful in providing virtual meeting support, we must integrate people, processes, and technology. Our cooperative visual hypermedia approach have used explicitly (visually) represented knowledge about the task at hand, the domain of discourse, and the set of participants involved to improve meeting process. It allows more flexibly move between and integrates different kinds of meetings or a series of meetings into a large business context. It makes contextual information persistently available when moving between different meeting processes. This makes it easier to integrate such environment with the enterprise-wide knowledge and project management systems as well as with the personal work environments of individual knowledge workers. This contrasts sharply with the currently available teleconferencing systems, which are stand-alone generic facilities with very limited capabilities.

Distributed meeting support involves a wide range of technical and social issues, and may use a wide range of communication and groupware technologies. In this work, we made a focus on distributed desktop-based meeting support for small group meetings. From our experience with the Dolphin system, this system can also be used for co-located meetings and the meetings with a mixture of co-located and distributed team members. For this potential, many useful features of the Dolphin system [25] could be re-established. We are also interested in the integration of mobile devices for remote meeting participants. As the current NetMeeting system has difficulties with firewalls, we are looking for alternative audio communication and application-sharing solutions that support http tunneling so as to have a better Web-based integration of our cooperative hypermedia based solutions. Currently one of our EXTERNAL project partners is performing evaluation on the XCHIPS system and other EXTERNAL tools, and we are cooperating with them for both feedback and findings.

Acknowledgments. We would like to thank our EXTERNAL project partners for feedback and discussions on the usage of the tools reported in the paper.

References

1. Akscyn, R. M., D. L. McCracken and E. A. Yoder, "KMS: A Distributed Hypermedia System for Managing Knowledge in Organizations," *Communications of the ACM*, v. 31, no. 7, July 1988, pp. 820-835.
2. Bentley, R., Horstmann, T., Sikkil, K., and Trevor J. Supporting Collaborative Information Sharing with the World Wide Web: The BSCW Shared Workspace System. In *Proceedings of the 4th International WWW Conference, Issue 1*, O'Reilly, Dec. 1995, pp. 63-74.
3. E. James Whitehead, Jr. WebDav, in *Proceedings of ACM Hypertext'01*, pp. 259, Aug. 14-18, 2001.
4. Ellis, C.A., Gibbs, S.J., and Rein, G.L. Groupware: Some issues and experiences. *Communications of the ACM* 34, 1 (Jan. 1991), 38-58.
5. Frank M. Shipman III, Haowei Hsieh, Preetam Maloor, J. Michael Moore, The Visual Knowledge Builder: A Second Generation Spatial Hypertext, in *Proceedings of ACM Hypertext'01*, pp 113-122, Aug. 14-18, 2001.
6. Greenberg, S., and Bohnet, R. Group Sketch: A multi-user sketchpad for geographically distributed small groups. In *Proceedings of Graphics Inte@aces'91* (June 5-7, Alberta, Canada), 1991, pp. 207-215.
7. Jessica Rubart, Joerg M. Haake, Daniel A. Tietze, Weigang Wang "Organizing shared enterprise workspaces using component-based cooperative hypermedia", in *Proceedings of ACM Hypertext2001*, pp 73-82, Aug. 14-18, 2001.
8. Kevin Wolf., "The Makings of a Good Meeting", 1994, available at <http://www.dcn.davis.ca.us/go/kjwolf/manual.html>
9. Lu, I., and Mantel, M. Idea management in a shared drawing tool. In *Proceedings of the European Conference on Computer-Supported Cooperative Work (EC-CSCW '91)*. (Sept. 24-27, Amsterdam), 1991, pp. 97-112.

10. Mantei, M., Baecker, R., Sellen, A., Buxton, W., Milligan, T., and Wellman, B. Experiences in the use of a media space. In Proceedings of the CHI '91 Conference (New Orleans, Louisiana), 1991.
11. Marilyn Manning, "Meetings Bloody Meetings", available at <http://www.mmanning.com/Articles/MeetingsBloodyMeetings.html>, Reprinted from March 1998 issue of Channel Magazine
12. Marshal, Catherine C and Shipman III, Frank M.: Spatial Hypertext and the Practice of Information Triage. In proc. of ACM Hypertext'97, pp : 124-133
13. McGuffin, L., and Olson, G. M. ShrEdit: A Shared Electronic Workspace. Technical Report No. 45, University of Michigan, Cognitive Sciences and Machine Intelligence Laboratory, 1992.
14. N.A. Streitz, Joerg Geissler, Joerg M. Haake, J. Hol DOLPHIN: Integrated Meeting Support across LiveBoards, Local and Remote Desktop Environments. In: Proceedings of the 1994 ACM CSCW'94, pp. 345-358, N.C., October 22-26, 1994
15. Nunamaker, J. F., et al. Electronic meeting systems to support group work. Communications of the ACM 34,7 (July 1991), 40-61.
16. Paul Kahn, Information Architecture: a New Discipline for Organizing Hypertext, Opening keynote speech at Hypertext'01, in Proceedings of ACM Hypertext'01, pp. 1, Aug. 14-18, 2001.
17. Rein, G. L., Ellis, C. A., rIBIS : a real-time group hypertext system, International Journal of Man-Machine Studies, 1991, volume 34, pages 349-367.
18. Reinert, O., Bucka-Lassen, D., Pederson, C.A., Nuernberg, P.J. CAOS: A Collaborative and Open Spatial Structure Service Component with Incremental Spatial Parsing. In Proceedings of Hypertext'97, ACM Press, Southamton, U.K., pp. 49-50
19. Roseman, M., and Greenberg, S. TeamRooms: Network Places for Collaboration. In Proceedings of CSCW'96, ACM Press, Cambridge, MA, 1996.
20. Stefii, M., et al. Beyond the chalkboard: Computer support for collaboration and problem solving in meetings. Communications of the ACM 30, 1 (Jan. 1987), 32-47.27.
21. Stotts, P.D. and Furuta, R. 1989. Programmable browsing semantics in trellis. In Proceedings Hypertext '89. Pp 27-42.
22. Weigang Wang "Team-and-Role-based Organizational Context and Access Control for Cooperative Hypermedia Environments", in Proceedings of ACM Hypertext'99, pp 37-46, Feb 23-25, 1999
23. Weigang Wang and Joerg M. Haake, "Flexible Coordination with Cooperative Hypermedia", Proceedings of ACM Hypertext'98, pp. 245-255, June, 1998