Updated and revised version previously published in Proceedings of ACM Conference on Hypertext (ECHT'92) (pp. 11–22). New York: ACM Press.

SEPIA: A Cooperative Hypermedia Authoring Environment

Norbert Streitz,
Jörg Haake, Jörg Hannemann, Andreas Lemke, Wolfgang Schuler, Helge Schütt, Manfred Thüring

IPSI – Integrated Publication and Information Systems Institute
GMD – German National Research Center for Information Technology
Dolivostr. 15, D – 64293 Darmstadt, Germany. e–mail: streitz@darmstadt.gmd.de

Abstract

This paper addresses two main areas: 1) research on computer–based support for cooperative authoring and 2) research on cooperative hypermedia systems. This is done by reporting about the design, development, and implementation of SEPIA and presenting results on dedicated and comprehensive authoring functionality addressing also the new rhetoric of hypermedia, a hypermedia data model with composites, persistent and shared data storage for hypermedia, and support for cooperative work, esp. cooperative writing.We start by identifying the challenge of hypermedia authoring and production which serves as the driving force for our development. Using interacting problem spaces as the vehicle for modelling the dynamic aspects of authoring, we arrive at a set of requirements answered by the concept of "activity spaces". The design of coherent hyperdocuments is facilitated by a "construction kit". Furthermore, we describe the extensions and modifications necessary to support multiple authors with the cooperative version of SEPIA. Based on the requirements, we develop a system architecture and report on the implementation of the system. We describe the basis for access to shared hyperdocuments, the activity space browsers, the integration of multimedia functionality (audio, graphics, pictures), and the integration of an audio and video conferencing system. Finally, we report on more recent developments and future work.

1 Introduction

This paper addresses two main areas: 1) research on computer–based support for authoring and cooperation of groups of authors and 2) research on cooperative hypermedia systems.The first part presents a cognitive framework viewing authoring as design problem solving followed by requirements for authoring systems derived on the basis of this framework. The second part describes the system architecture and the implementation of SEPIA, a cooperative hypermedia system, meeting most of these requirements. SEPIA is a realization of our approach that hypermedia is not only the content or the subject matter but also a medium for computer-supported cooperative work in general (Streitz, 1994 a), but especially for cooperative authoring.

In his "Seven Issues: Revisited", Halasz (1991) mentions a 'broader vision of what constitutes the world of hypermedia'. He proposes a five-level system architecture and distinguishes: data storage substrate, data models, navigational facilities, applications, issues of situated use. Most of these aspects are treated in the context of the design, development, and implementation of the SEPIA cooperative hypermedia authoring environment. In this paper, we will report about this ongoing research resulting in a hypermedia system which addresses four of these five levels. It provides results on persistent and shared data storage, hypermedia data model with composites, dedicated and comprehensive authoring functionality at the application level, support for a new rhetoric and for cooperative work at the situated use level.

The idea of SEPIA (*S*tructured *E*licitation and *P*rocessing of *I*deas for *A*uthoring) and its basic design principles were first described in Streitz et al. (1989). We like to note that the point of view that authors of hypermedia documents need specialized support is still or even more valid than at the time of our previous publication in 1989. We also learned the lesson that one cannot develop authoring tools without being aware of what a presentation environment requires. This is reflected in our R&D strategy which addresses the cognitive processes, the product, and the social aspects of the authoring activity. Figure 1 shows the relationship of the activity under investigation, the theoretical basis, and the resulting components of SEPIA. Paying attention to the process aspect requires to develop and refine a model of the cognitive processes of writing and to transform these results into requirements, as e.g. in our activity space concept. Looking at hyperdocuments as a product with features of a new rhetoric (Thüring et al., 1991, 1995) results in requirements for a corresponding functionality, as e.g. our construction kit in the rhetorical space. To get valid requirements, we built a large hyperdocument based on the SEPIA presentation interface (SPI) as a separate reading environment in order to test our assumptions about a new rhetoric for hypermedia (Hannemann et al., 1992, Thüring et al., 1995). Considering that most large and complex documents are prepared by a team (Streitz, 1995), social cooperation models had to be defined, and SEPIA had to be extended from a single-author to a multiple-author environment by providing corresponding cooperation modes (Haake & Wilson, 1992). Thus, detailed knowledge about the process, the product, and the social situation played equally important roles in the development of our user-oriented and task-driven authoring environment.

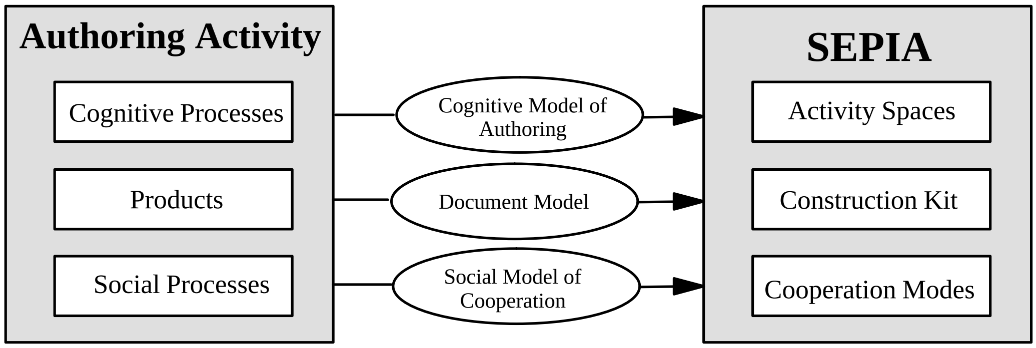

Figure 1: Research and development strategy for SEPIA

2 The Challenge of Hypermedia Authoring

Readers as well as authors have to struggle with a variety of problems arising from the net-like character of hypertext.

The *reader* of a hyperdocument is typically confronted with two difficulties. The first one is known as the navigation problem: Readers often *"get lost in hyperspace"* (Conklin, 1987). Most of the research on reading hypertext has solely concentrated on the navigation problem thus overlooking a second difficulty: Many readers have trouble to *comprehend* a hyperdocument, i.e., they often fail to grasp its overall structure or to understand the semantics of links. Disorientation and deficient comprehension probably have the same cause: readers are impeded in forming a *coherent* mental representation of the document. As a consequence, more and more readers complain about the low quality of hyperdocuments.

To create hyperdocuments of high quality, the *author* must be aware of his readers' problems and view them as problems caused by him – at least to a certain degree. Especially, he is responsible for designing hypertext structures and presentation formats which increase the coherence of his document and support efficient navigation. But this is not an easy task. In contrast to writers of linear documents, authors of hyperdocuments have no guidelines telling them what their product should look like. Many rhetorical decisions must be made without the security of widely accepted conventions. Since these decisions entail activities supplementary to the processes of writing a linear text, such activities are often regarded as *cognitive overhead* (Conklin, 1987).

The lack of rhetorical guidelines and cognitive overhead complicates the authoring of hyperdocuments and contributes to the low quality to be observed in many cases. To overcome this unsatisfactory state, the *developers* of hypertext systems must be more aware of reader- and author-specific problems and the construction of writing tools must be based on a sound theoretical foundation. Applying the basic principle of *cognitive compatibility* (Streitz, 1987), we have translated this insight into the requirement that authoring systems which are intended to give appropriate support must be cognitively compatible to authoring activities (Streitz, et al., 1989).

2.1 A Cognitive Framework: Authoring as Design Problem Solving

Based on an analysis of the cognitive processes of writing and the features of the authoring situation, we have characterized writing as a design activity (Hannemann et al., 1990). The interdependencies of extensive planning, production and revision activities are characteristic for the writing process and lead to both, an external product – the text – and an internal product – a new knowledge structure. Just as readers may find it difficult to explore the hyperspace, authors find it difficult to explore the complex design space. Helping an author 'travelling' through this space, the development of an authoring environment must rely on three main features of every design process:

- Design is a complex *problem solving process*, which consists of different *subproblems*. These problems are solved by specific activities which are *opportunistic*, i.e., they strongly interact and build on each others' results.
- Design is the *construction of an artifact* which has to fulfil specific criteria and for which the designer needs adequate *building blocks* to compose the artifact.
- Design usually is a *social process* that involves a group of individuals. Therefore, facilities which support *cooperation* should be incorporated into an authoring environment.

Now, we describe the implications of these aspects and derive requirements for the development of SEPIA.

2.2 Supporting the Design Process: Activity Spaces for Hypermedia Authoring

Using results of research on writing, we have identified three closely related subproblems which an author must solve to produce a document : the content problem, the rhetorical problem, and the planning problem. According to Newell (1980), the mental representation of these three problems can be described in terms of separate but *interacting problem spaces* formed by different constraints, design objects and operations in which different knowledge sources are brought to bear. Applying the principle of 'cognitive compatibility', we use this decomposition of the design space into subspaces as a basis for dedicated requirements of components of the authoring environment. These (cognitive) problem spaces are "matched" in the SEPIA system by corresponding *activity spaces.* Each activity space provides specific design objects and operations appropriate to facilitate the author's activities when working on the above subproblems:

- the content space,
- the rhetorical space, and
- the planning space.

Since argumentation is a crucial cognitive activity which plays an important role in writing for a large number of document types, we supplemented these three spaces by a fourth space called

- argumentation space.

To support the *construction of artifacts* SEPIA provides a special 'construction kit' wich is integrated in the rhetorical space (for more details see Thüring et al., 1991, 1995).

2.3 Supporting the Social Process: From Single to Multiple Authors

A main feature of the authoring process is that it involves in many cases more than one person. This is especially true for hypermedia (Streitz, 1995). The design of SEPIA has to reflect this by providing support for the *cooperation of authors working in groups*. This involves the following activities.
First of all, authors access and modify *shared hyperdocuments* concurrently. The environment should allow a maximum of concurrent activities by the authors whenever they work on different parts of the document. Authors working on the same part of the document should be prevented from accidentally destroying each other's work. Group authoring occurs in different modes of collaboration that we label: *individual*, *loosely-coupled*, and *tightly-coupled* work. The modes differ in the *level of awareness* each author has of the activities of the coauthors. In *individual work***,** a single author manipulates a task-specific cluster of nodes and links. Even though the author works individually, there is a need to collaborate with the coauthors asynchronously, for instance, through an *annotation facility***.** In *loosely-coupled work*, several coauthors working on the same subtask manipulate the same cluster. In this mode, they need to be aware of each others presence and activities. In *tightly-coupled work*, authors cooperate and coordinate their work in synchronous conference-like "meetings." In this mode, authors should be provided with a WYSIWIS–functionality (WYSIWIS–What You See Is What I See) and additional channels for meta-communication. Cooperative writing proceeds by shifting between these three collaboration modes. Due to the opportunistic nature of cooperative writing, one cannot foresee the sequence of the collaboration modes. Therefore, *smooth transitions between the modes* must be supported.

2.4 Summary of Requirements

To support both individual and cooperative writing of hypermedia documents, SEPIA should therefore meet the following requirements. It should

(R1) support activity spaces for hypermedia authoring, i.e.

- provide task-specific objects and operations,
- provide views on hyperdocument structure (network level) and content (node level),
- provide a cognitively compatible user-interface,

- support exchange and cross-referencing of objects across activity spaces,
- provide a hypermedia data model which is tailorable to activity spaces and tasks,
- provide persistent storage for structure, content, and view information,
- support multimedia data (text, sound, voice, graphics, pictures, etc.).

(R2) support versioning of hyperdocuments.

(R3) support distributed authoring of hyperdocuments including access to a shared hyperdocument database.

(R4) support shared workspaces at the network and the node level. These

- are shared browsers for activity spaces, resp. composite nodes in general.
- support different collaborative modes (individual, loosely coupled, tightly coupled),
- allow for smooth transition between modes.

(R5) support additional channels for meta-communication (audio, video conferencing).

3 Functionality and User-Interface

Within each activity space there are two levels of operations: the *network level* (navigation in the graphical browser and actively editing the network) and the *node level* (reading and editing content).

3.1 Authoring at the Network Level

3.1.1 Activity Space Functionality

Figure 2 shows a screendump of open activity space browsers. They were opened by clicking on their initials (P, C, A, R) in the 'launcher' of the 'project' "Telecoop". (The example was created in the context of the POLIKOM–project on telecooperation support between Bonn and Berlin, Hoschka et al, 1992). Users can browse in each space by activating nodes and links and scrolling if the current window does not contain all objects. A 'roaming box' (upper left corner) shows an actively manipulable presentation of the whole space (resp. composite node). Each space provides space-specific functionality (typed nodes, links, operations) available for every author independent from his cooperative work mode.

The design objects and operations of the **content space** are dedicated to facilitate the development of a domain model. For this purpose, SEPIA provides the structuring facility of hypertext to support idea dumping, their grouping in topic related clusters by composite nodes and connecting them via typed links. This can also involve access to background material either from internal (e.g., previous documents) or external sources (e.g., querying a data base).

In the **rhetorical space**, the author creates the reader-oriented, final document. This final product can be both a conventional, linear text or a hyperdocument, formed by a typical network of nodes and links. Both document types constitute a scale ranging from strictly linear to strictly non-linear documents. Notice that hyperdocuments can vary in the degree of their linearity between these two endpoints. Nevertheless, they all should satisfy one major requirement: In order to support comprehension and navigation on behalf of the readers, they must appear as *coherent* entities. Therefore, the rhetorical space provides a special 'construction kit' based on the concept of coherence consisting of design objects that are explicitly tailored to the requirements of designing artifacts (Thüring et al., 1991, 1995).

In the **planning space**, an author has the opportunity to externalize his writing plans, resp. goals, to construct issues to be concerned with in the document, and to establish an agenda for the authoring activity. Consequently, this space serves as a meta space for coordinating the activities in the other three spaces and for controlling the progress of the design process.

For the development of an issue structure, SEPIA provides a set of dedicated nodes and links. We use a modification of the IBIS method (Kunz & Rittel, 1970) by extending the issue concept and introducing a new principle for linking issues (Schuler & Smith, 1990). In addition, the planning space is linked tightly to the argumentation space. 'Positions' which are formulated as 'answers' to issues in the planning space are transformed and recreated as 'claims' in the argumentation space prompting the author for providing supporting arguments (example in fig. 2: 'Horizontal distribution').

The **argumentation space** supports the development of an argumentation structure by providing appropriate design objects and operations based on our extension of the argumentation schema developed by Toulmin (1959). Using the argumentation space, the author can elaborate an argumentation by generating support or objections at different levels, by formulating contradictions and by constructing argumentative chains (for details see Streitz et al., 1989).

When 'travelling through activity spaces', the author does not need to follow a predetermined route. At every point in the authoring process, he can decide which subspace to use next. To support the high flexibility for interaction and smooth transformation of knowledge between the activity spaces, SEPIA allows automatic transfer of design objects between specified spaces,

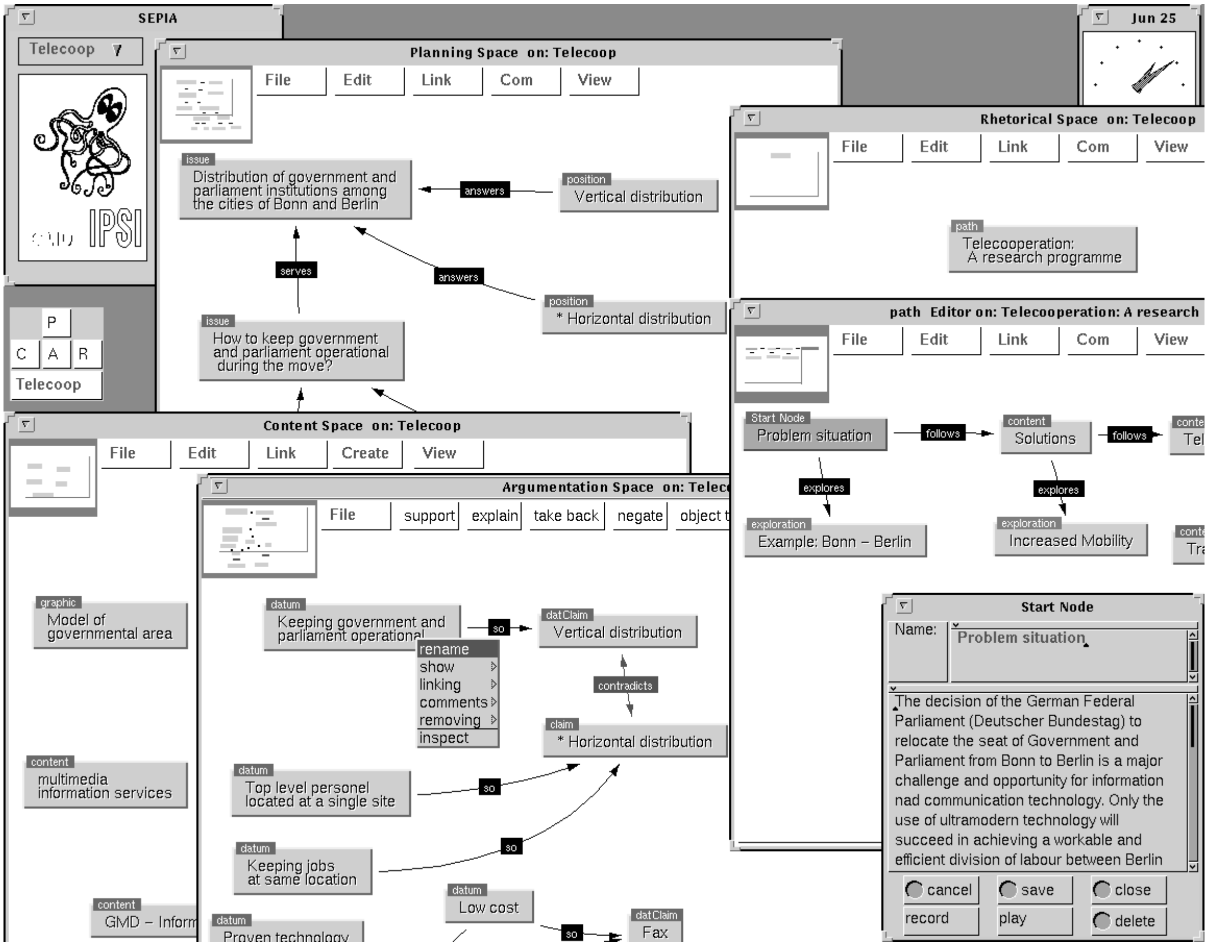

Figure 2: User-Interface of SEPIA

their reuse, and the indication and control of references between activity spaces.

3.1.2 Multiple Authors

Figure 3 shows the user-interface of an author who is in different composites in different cooperation modes.
When several coauthors work on the same task (i.e., each of them has an active browser on the corresponding composite node), the respective browsers initially are in *loosely coupled mode* (Planning space in fig. 3). Authors are made aware of each other via (1) a list of all concurrent users displayed in the resp. browsers (e.g. schuler, haake) (2) highlighting of objects locked by other users, and (3) a relaxed WYSIWIS view. Actions affecting the view of the node are private, but manipulations of objects in the node become visible immediately to all other browsers if they affect the currently visible area. Locking at the data base level is used to prevent coauthors from simultaneously modifying the same object.

In *tightly coupled mode*, the coupled browsers display a WYSIWIS-view on the composite node's content. For an example see the Content Space in figure 3 showing two tightly coupled users (schuler, haake) and one additional loosely coupled user (hannemann). In addition to the functionality of the loosely coupled mode, scrolling and resizing events are immediately broadcasted to all tightly coupled browsers.
Awareness of the coauthors' activities is a prerequisite for smooth ad hoc *transitions* from one mode of collaboration to another. Currently, the transition from individual work to loosely-coupled work is triggered automatically when a second author opens a composite node already "occupied" by the first author. This is indicated by a "door bell" sound on both workstations and the change of the user list. Being in loosely-coupled mode, authors might want to join for a tightly-coupled session. To start a tightly-coupled session, one coauthor selects all or a subset of those coauthors currently in the same node and invites them to participate in the session. The system asks each of them to confirm. The browsers of

those coauthors who confirmed are shifted into tightly-coupled mode. Authors can exit a tightly-coupled session either by closing the composite node or by returning to loosely-coupled mode (Haake & Wilson, 1992).

3.2 Node Level

Beyond the structural aspects at the network level, hyperdocuments are very much characterized by the type of media which are used. All atomic (content) nodes can carry *multimedia information* including text, graphics, pictures, and sound. They can also be annotated by multimedia nodes. We are also working on the integration of digital video as content of a node. This use of multimedia has to be distinguished from using it for communication purposes as, e.g., in audio and video conferencing systems.

Cooperative use of hypermedia requires joint viewing and editing of the content of a node. This is facilitated by integrating *WScrawl* (Wilson, 1992), a group-aware, color, pixel-oriented, shared drawing tool we have also used as part of another desktop conferencing scenario (Lemke et al., 1992). Here, it is used for displaying and editing the graphics and picture content of nodes (e.g., the picture of the "Model of the governmental area" displayed for the users "schuler" and "haake" in fig. 3). Each drawing action is immediately visible on all connected displays. Users can import arbitrary information from their screens (even outside of WScrawl or SEPIA windows) using the 'SuperSelect' facility and show it to everybody currently sharing the view in WScrawl.

3.3 Meta-Communication Channels

Having shared browsers is only one way of supporting synchronous remote cooperation. As indicated by the 'interactive communication model' in CoLab (Tatar et al., 1991), additional communication channels are required. SEPIA provides a digital audio channel for audio-only conferencing as well as an analog audio/video conferencing device enabling up to four coauthors to see and talk to each other (see Collaboration menu in fig. 3). In addition, SEPIA supports gesturing by providing concurrent telepointing for each tightly coupled user (e.g. haake) at the network level and within WScrawl at the node level. Each telepointer displays the name of its user. Furthermore WScrawl serves as a common scratch

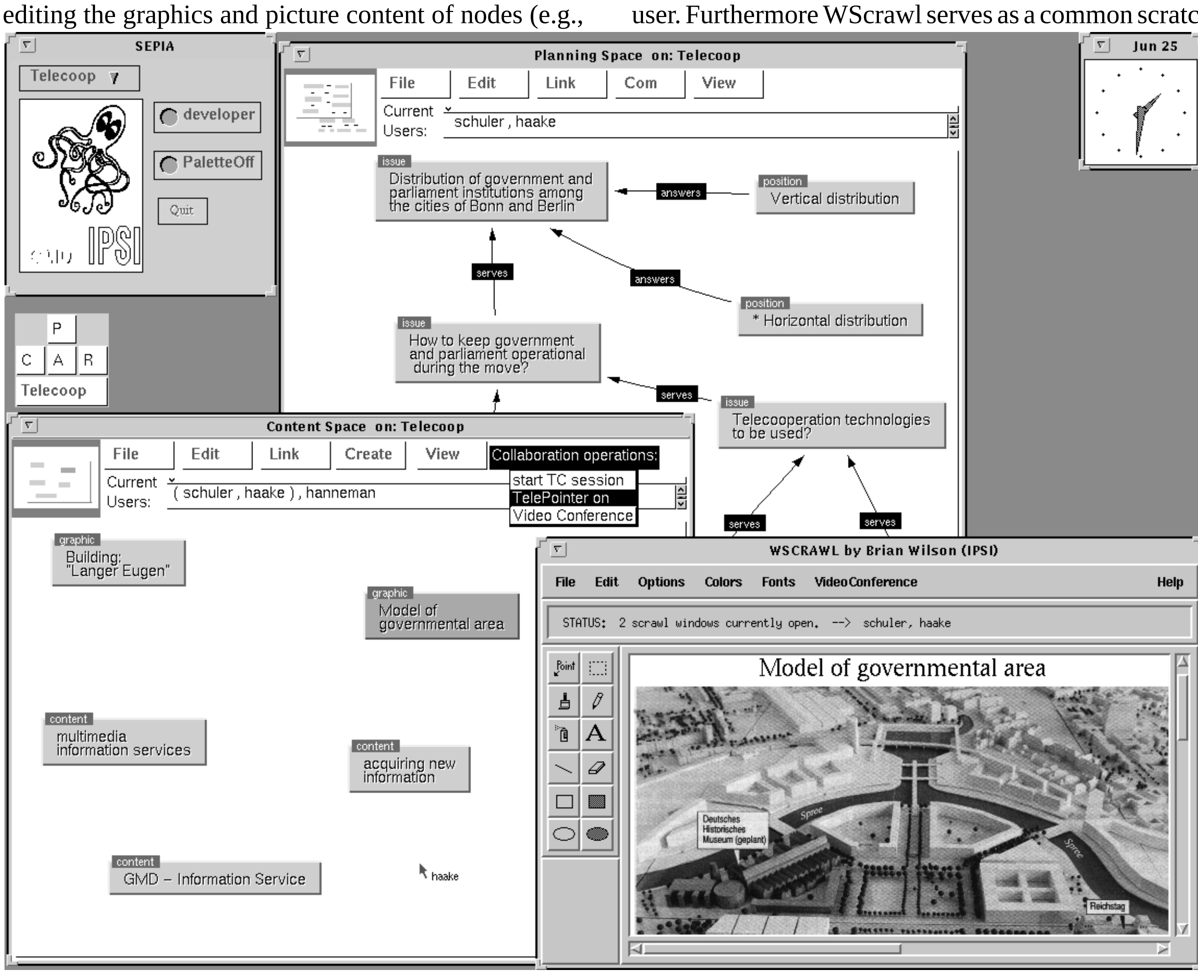

Figure 3: User-Interface of cooperative SEPIA

space for coauthors in tightly coupled sessions. Groups of authors can take meetings notes which are available immediately to any group member and which can be attached to the hyperdocument if desired.

4 System Architecture and Implementation

First, we describe the single author system before we discuss the implementation of the cooperative system.

4.1 The Single-Author System

In order to meet the requirements R1 and R2, we chose the architecture shown in figure 4 which we discuss in a bottom-up fashion.

We distinguish between an application module (SEPIA's activity spaces) and an object management module. In the latter, the hyperdocument data (nodes, links, and composites, together with their attributes) are handled by the hypermedia engine *HyperBase* (Schütt & Streitz, 1990) which provides a persistent storage and retrieval mechanism for hyperdocuments. These are stored in a *hyperdocument database***.** *HyperBase* was extended to *CHS* (Cooperative Hypermedia Server, cf. section 4.2).

To support versioning (R2), we integrated *CoVer*, a contextual version server which provides basic versioning concepts to the activity space browsers. The distinguishing feature of CoVer is that it not only maintains versions of individual objects but it also maintains the task context in which versions are created. See Haake (1992, 1994) and Haake & Haake (1993) for details.

All hypermedia objects are implemented as refinements of a generic data model which was specified using the *SFK frame system* developed at IPSI (Fischer & Rostek, 1995). A frame-based approach is well suited for the modelling of typed hypermedia structures because it allows the declarative specification of constraints which can be checked at runtime (e.g., validity of link sources and destinations), it is easily extendible (tailorability), and it supports transactions within the authoring environment.

Our generic data model combines hypertext constructs with object-oriented frame-based representations similar to Aquanet (Marshall et al., 1991) or MacWeb (Nanard & Nanard, 1991). All its entities are represented as frame objects with single inheritance. Their named and typed slots carry content, structure, and system information, as well as attributes. The basic data model objects are typed nodes and links (see fig. 5), where the types are realized as frame classes. The content slot of the nodes contains the hypermedia's primitive data types (e.g. text, image, sound). Links are also typed first-class objects. They represent relationships between SEPIA objects. Their type definition includes their constraint information.

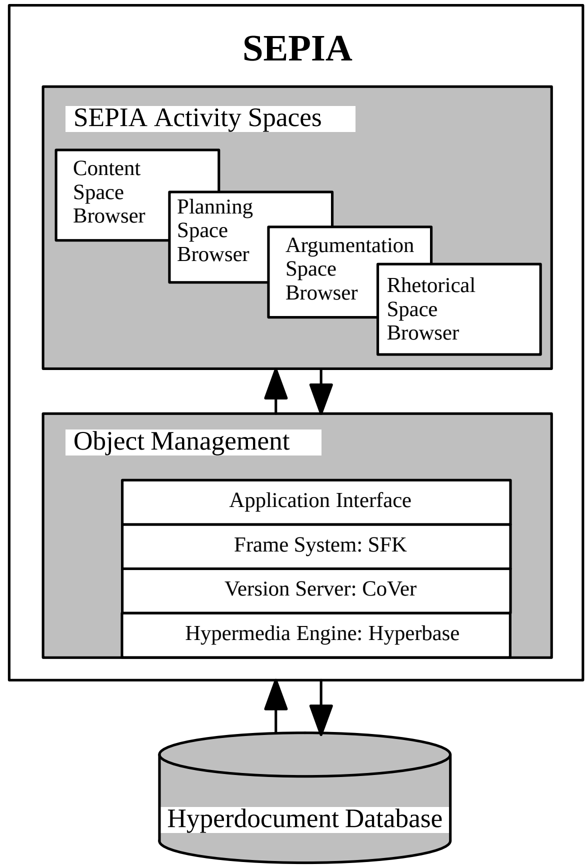

Figure 4: System architecture of SEPIA for single authors

Source and destination objects of a link are link anchor objects which are associated to the basic objects of the link relation. Anchors are conceived as logical and not geometrical entities bound to a portion of text or picture. The *application interface* (see fig. 4) defines the mapping of the hypermedia data model to the data model of *HyperBase.*

SEPIA's basic structuring means are *composite nodes* which contain a partially ordered set of basic objects (nodes and links). Composite nodes are used to represent subgraphs of the hypermedia network. Activity spaces and folders used in an activity space are implemented as composite nodes. Each activity space uses an application-dependent subset of the node and link types. *Activity space browsers* provide activity-specific views on the hyperdocument. These views provide access to task-specific objects and operations. Activity space browsers can exchange hypermedia objects or can create reference links to objects of other activity spaces. In a hypermedia environment it is important to support persistent view information because structuring the layout of a hypermedia network is an additional important feature for authors.

For every data object (node, link, composite node) to be displayed in an activity space browser, SEPIA uses a special persistent *container object* which contains its view information. Thus, all display information (e.g. position, icon, style, size in a graphical net browser) of a data object to be shown in a browser is stored in a special container object which is typed according to the data object. As a result, one data object can be connected with different container objects and therefore be displayed differently in different contexts via different container objects. Example: A 'position' in the planning space can be displayed in its role as a 'claim' in the argumentation space at the same time.

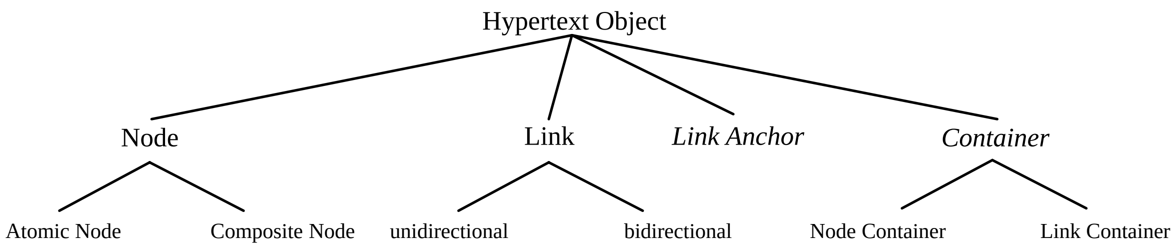

Figure 5: SEPIA Data Model Hierarchy
Basic data objects are shown in normal letters, view objects in italic.

4.2 The Multi-Author System

In order to meet the requirements for cooperative authoring (R3 - R5), we extended the architecture as follows (see figure 6).

The *cooperative hypermedia server CHS* provides shared access to hypermedia documents. We explored several ways of implementing shared hypermedia servers, such as building CHS on top of a multi–user DBMS or using a file–system based approach. In the DBMS approach, CHS can exploit the transaction management facilities of the underlying database system for concurrency control and recovery. It captures deadlock and livelock situations and ensures that the hyperdocument database is always in a consistent state. The adopted client-server architecture enables multiple clients to access the same database server in a distributed computing environment (R3). In order to support collaboration, CHS maintains a list of users which are currently logged into the system. The data model of CHS has been extended by locks which can be assigned to objects. The *application interface* now not only defines a mapping from the clients' data model to the data model of CHS, but it also defines policies for transfering data between a client and the shared database (more details are found in Schütt & Haake,1993).

4.2.1 Shared Workspaces at the Network Level

Shared workspaces (R4) have been realized through application interfaces and browsers which exchange update information. In SEPIA, all changes are immediately stored in the shared database. In addition to this, change notifications are broadcasted among SEPIA clients in two ways:

First, the *application interface* broadcasts change notifications of hypermedia objects stored in the database to ensure that all clients use the same state of the shared objects. This feature is used to realize the loosely coupled mode of activity space browsers. A *broadcast server* is connected to all SEPIA clients which broadcasts change notifications among the clients. Every SEPIA client includes a *broadcast listener process* which waits for change notifications from the broadcast server.

Second, *activity space browsers* which are in tightly coupled mode communicate directly with each other. They exchange messages synchronizing scrolling, resizing, and telepointing. These messages are received by a local *communication handler* which is associated with each browser. Each browser has a session object attached to it which keeps track of cooperation modes and current users.

4.2.2 Shared Workspaces at the Node Level

Sharing of information is also available at the node level. In SEPIA, we use the WScrawl shared drawing tool for that purpose. Opening a node with graphical contents starts WScrawl which provides an arbitrary number of authors with a shared whiteboard every author can write on. WScrawl uses the X window server to synchronize event handling and to exchange data among participating authors. When WScrawl is called from a browser which is in tightly coupled mode, it is also started in tightly coupled mode and displays the content of the node to be shared automatically on all users' workstations which are currently in tightly coupled with this browser.

4.2.3 Additional Communication Channels

In order to provide additional communication channels (R5), we are following two technical routes: analog and digital. Digital audio is automatically set up as a bidirectional communication line – over the ethernet – among two tightly coupled users in a session. There are appropriate *mike* and *speaker processes* installed at the beginning of a tightly coupled session at each workstation, which send and receive audio packets to or from the remote partners. Only one tightly coupled session at a time can use the audio communication facility due to the limited mike and speaker resources. When finishing a tightly coupled session the corresponding audio processes are automatically shut down.

On the other hand, we integrated a video conference system installed at IPSI into the cooperative SEPIA clients. As part of the SEPIA interface, a user can ask the *video router* to provide an analog video and audio connection between him / her and remote partners. Initially, the video conference was limited to connections of two people, but by using now a cross split video switch, simultaneous video conferencing is available for up to four people at a time.

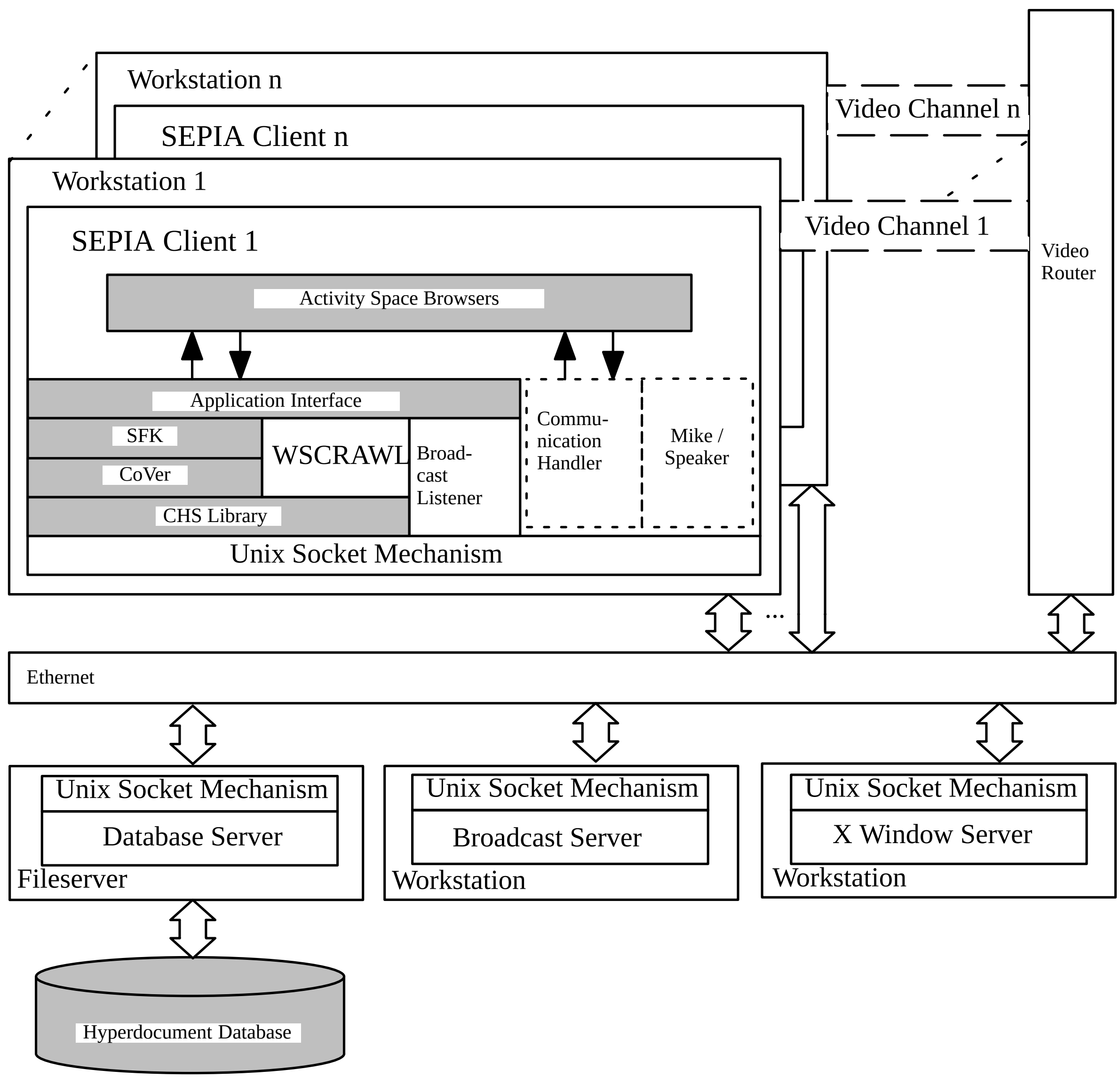

Figure 6: System Architecture for Cooperative SEPIA

Shaded areas represent components which already exist in the SEPIA system for single authors. The communication handler and the mike / speaker processes are represented as dotted lines because they are created on demand only.

4.3 Implementation Details

The cooperative SEPIA clients are implemented in Smalltalk–80 (Parc Place VisualWorks 1.0) on SUN Sparc–10 workstations running SUN OS 4.1.3 Unix and the OpenWindows 3.0 window system. CHS is implemented in C on top of the relational DBMS Sybase. The audio communication feature uses the Netfone software, release 1 (Walker, 1991) written in C. The video server and WScrawl are implemented in C. WScrawl uses an X server to broadcast information. Interprocess communication is implemented by using standard UNIX sockets.

5 Conclusions and Future Work

We now discuss the innovative aspects of SEPIA on the following three dimensions of a design process and compare it with other approaches having similar goals:

1. User-oriented and task-driven system design resulting in support for different subtasks of the authoring activity
2. Support for the special requirements of hyperdocument production
3. Support for different modes of cooperating authors

5.1 Dedicated Support for Different Subtasks of the Authoring Activity

With respect to the first dimension, we like to state that there are only few attempts to address the problem of dedicated support for hypermedia authoring. Many systems focus primarily on providing presentation and reading environments. Similar to our approach – in terms of cognitive modelling – is the development of the Writing Environment (WE) (Smith et al., 1987). It supports the creation of a network of ideas (network mode) to be transformed in a hierarchical document structure which is then displayed and can be edited in a linear mode. The network mode exhibits a node-link structure but does not offer different types of nodes and links. The final document is a linear document. While WE did not support multimedia components nor collaborative writing, the successor system ABC (Artifact–Based Collaboration) (Smith & Smith, 1991) provides support for cooperative work. StorySpace (Joyce, 1991) provides an interesting interface metapher for authoring hypertext. Other relevant approaches for the support of authoring are the MUCH system (Rada et al., 1992, 1993) and the RICH system (Wang & Rada, 1995).

This overview would be incomplete without mentioning systems for argumentation support: gIBIS (Conklin & Begeman, 1988), PHIDIAS (McCall et al., 1990), EUCLID (Smolensky et al., 1987), Toulmin in NoteCards (Marshall, 1989). They have in common that they use either the IBIS-approach or the Toulmin model of argumentation. Aquanet (Marshall et al., 1991) provides schemata to create Toulmin or IBIS-structures. Although these systems offer dedicated support for one aspect of the authoring activity (argumentation, knowledge structuring) they lack support for the other processes we have identified and realized in the four activity spaces of SEPIA.

In summary of the first dimension, we can state that SEPIA is special in so far as it provides support for a wide range of hypermedia authoring activities. It is theoretically well grounded and reflects the theory via the provision of dedicated node and link types and corresponding operations in each space. In addition, it offers interspace linking possibilities and the reuse of hypermedia structures and content across subtasks.

5.2 Dedicated Support for Hyperdocument Production

As introduced in chapter 2, the production of hyperdocuments requires concepts and methodologies with respect to the final product. Authors need support on different levels of hypermedia networks. Conceptually related to our construction kit (Thüring et al. 1991, 1995), is HDM – the hypermedia design model – proposed by Garzotto et al. (1991, 1993). It utilizes a schema approach for efficient high level structuring of large applications and subsequent instantiation. Another attempt to provide high level concepts to the author was proposed by Smith Catlin et al. (1991). They extended Intermedia with templates realized as a set of pre-linked documents which can contain both content and formatting information. But they do not explicitly address the problem of coherence. A related approach for providing high–level overviews is the work by You & Rada (1994) on outline manipulation in the context of the hypertext system MUCH where they exploit the potential of hypertext for creating alternative views in a dynamic fashion.

5.3 Support for Different Modes of Cooperating Authors

SEPIA realizes two roles which from our point of view (Streitz, 1994 a) hypermedia can and should play for cooperative work: 1) Hypermedia constitute the content of cooperation and 2) Hypermedia provides a base technology and represents a medium for facilitating cooperation. This is in line with the observation of Halasz (1988, p. 848): “Hypermedia is a natural medium for supporting collaborative work.” Comparing SEPIA with other systems on this "cooperative" dimension yields the following observations.

The GROVE group text editor (Ellis & Gibbs, 1989; Ellis et al., 1991) uses local editors and replicated documents together with a central coordinator serializing all editing operations. SEPIA is geared towards hyperdocument authoring and synchronization is done using transactions and locking of objects in the database. Instead of broadcasting *operations* which are difficult to sequentialize, SEPIA broadcasts *update notifications* which need not be sequentialized by the clients. The rIBIS system (Rein & Ellis, 1991) is based on a central server architecture providing one TC session per hyperdocument and supporting only one mouse – a group mouse – within a TC session. SEPIA is implemented following the replicated architecture approach providing multiple TC sessions per composite node and supporting a private mouse for each coauthor. Switching between cooperative modes is very smooth in SEPIA. While Dewan & Choudhary's (1991) collaboration support environment requires users to tailor the coupling behaviour to their needs, cooperative SEPIA relieves users from constructing a specific coupling behaviour. Furthermore, SEPIA provides concurrency control and maintains dynamic sessions. Aquanet (Marshall et al., 1991) follows also the replicated architecture approach but does not support synchronous cooperation in terms of shared views, telepointers, and audio communication. The PREP-editor (Neuwirth et al., 1990, 1994) supports asynchronous collaboration addressing cooperative aspects of writing in terms of common planning and annotation activities and flexible diffing (Neuwirth et al, 1992, and this volume). Although not developed as a hypertext system in the first place, it makes use of some hypertext features in an interesting fashion as, e.g. separate but linked columns for each author and for annotations.

5.4 New Developments and Future Work

In this part, we reflect on our previous experiences with SEPIA, provide information on new developments of the system since the first publication of the original paper, and point to future work. In our early uses of the prototype system, we have been concerned with (argumentative) proposal writing, project planning, recording user feedback in SEPIA itself, and replicating the task of a science journalist writing an article on a scientific debate. Our experiences show that the system is appropriate for these tasks. In order to test SEPIA in *new application domains* we followed several routes.

First, we experimented with adapting existing activity spaces to new task contexts. Examples are cooperative decision making and capturing design rationale. This was achieved by focussing especially on the planning space and the argumentation space and by modifying the types of nodes and links appropriately.

Another approach was taken by using the tailorability of the activity space framework and defining new application-specific spaces. An example is the so called *MuSE–SEPIA* (Haake et al., 1994) which was developed in the MuSE–project (Multimedia Systems Engineering). This project aims at providing a computer-based environment for the engineering of complex technical systems (e.g., trucks, aircrafts). The system comprises both the validation of the system model (before it is actually built) and support for modeling techniques required by the designer. For this application, we developed new spaces, e.g., a modeling space and a validation space with new types of nodes and links derived from the requirements of the system engineers.

Quite a different route has been taken by developing the DOLPHIN system (Streitz et al., 1994) which provides support for a range of collaborative activities, e.g., face-to-face meetings in electronic meeting rooms, which were *outside the original scope of SEPIA*. But DOLPHIN can be used in a complimentary way because of its full compatibility to SEPIA. We will come back to this later on.

Finally, we have developed a number of *components as extensions or enhancements* starting with the original SEPIA system but to be used in most of the new, sometimes rather application-specific variations of SEPIA. This includes the *reading environment SPI (SEPIA Presentation Interface), a linearizer tool, object–oriented database support, and versioning*.

Although SEPIA is a hypermedia authoring environment, we have to acknowledge that we still live in the context of traditional linear documents and printed paper output. Therefore, we have developed a *linearizer tool* which transforms a hyperdocument in the rhetorical space into a linear document (Knopik & Bapat, 1994, Lakoumentas, 1995). The overall strategy is composed of substrategies at four different levels (graph traversal, insertion into the linear list of nodes, handling substructures, creation of export format). In our implementation, we allow the user to select these substrategies. For example, we offer ASCII, Interleaf, and HTML as export formats. Conversely, an Interleaf document with appropriate markup can be read in and converted to a hyperdocument which is based on our notion of paths. We will also extend the transformation process to produce documents which conform to the SGML standard (ISO, 1986). Closely related to this are enhancements in the rhetorical space building on the idea of the construction kit (Thüring, et. al, 1995). This includes to provide special predefined document types for later use. The definitions will be based on SGML or HyTime (Newcomb et al., 1991; DeRose & Durand, 1994).

Other extensions we have developed are a *graphical path editor* and a *reading evironment.* The path editor allows the definition and modification of path conditions bound to nodes and links in the rhetorical space. A path interpreter is used to evaluate such conditions (which are based on a history of visited nodes and links) at run time. The reading environment provides the author with previews of the final presentation (Hannemann et al., 1992). Related work on the design of user-interfaces for hypermedia applications is described in Schuler et al. (1995).

As mentioned before, we are working on *enhancements for the DBMS support*. This is part of HyperStorM, a joint project of two research divisions within IPSI. It is concerned with the integration of the SEPIA system and the VODAK system which is a distributed object-oriented DBMS developed at GMD-IPSI (Klas et al., 1990). HyperStorM develops an extendable object–oriented hypermedia engine which supports the specification of application semantics as object classes in the hypermedia engine (Wäsch & Aberer, 1995. There is also related work at IPSI on storing HyTime documents in an object-oriented database (Böhm & Aberer, 1994).

Furthermore, we have been working on the integration of *versioning support* into SEPIA. This is based on the contextual version server CoVer (Haake, 1992, 1994). Here, we address not only the object management but with equal importance the user-interface aspects of how to present versions of hypermedia objects to the user. A first attempt is described in Haake & Haake (1993). Versioning is being used in several application situations. While it is useful in managing different stages and intermediary products of the authoring activity for a single author it becomes more relevant and even crucial in cooperative writing situations. For example, the history cards in NoteCards (Irish & Trigg, 1989) list log of updates. But beyond this one is also interested in the content of previous versions, their properties, and their relationships to the current version. We are also looking at other applications (DOLPHIN) with a need for versioning support, e.g., for tracking the development of ideas and discussions in group meetings including the audio and video coverage of the sequence of events and their correspondences in the hypermedia structures.

While the developments mentioned before are very relevant, the *most important change took place with respect to CSCW support*. Here, we have extended our view to a more global perspective (Streitz, 1994 a, Streitz 1994 b). A comprehensive view of the overall collaboration scenario of teams of authors creating large-size documents, or of cooperating work groups in general, reveals the central and important role of face-to-face meetings, e.g., for initial planning and brainstorming activities and, later on, for review processes and editorial decisions. Thus, we built an electronic meeting room, the OCEAN-Lab, where each participant can use a notepad computer mounted in the table. The computers are connected to each other by a local network and with a large public display which is also used as an interactive electronic whiteboard (Streitz et al, 1994). It was clear that these activities require a different set of tools and especially different structures than those provided by SEPIA. Similar to the use of traditional blackboards/ whiteboards in meeting rooms, people need possibilities for communicating their ideas via handdrawn sketches and scribbles, making annotations, etc. This requires a wide range of structures from more informal in early phases to more formal structures in later phases (Haake, Neuwirth & Streitz, 1994). This functionality is now provided by the DOLPHIN system (Streitz et al., 1994) where one can write with a cordless pen, use gestures for frequently used operations. One can also transform/ develop these informal structures into more formal hypermedia structures with nodes and links. All structures can exist in coexistence with each other on the same “page”, as part of different composites, and at different levels of these non–linear networks. In order to obtain feedback on our design decisions, we ran an evaluation experiment. Details can be found in Mark et al. (1995). DOLPHIN is based on an improved version of the cooperative systems architecture of SEPIA. It employs a very different interaction paradigm (including gesture recognition for frequently used operations) and accomodates different types of objects as, e.g., handwritten scribbles which can be turned into hypermedia nodes.

In the case of larger groups of authors cooperating over longer periods of time, coordination becomes crucial. In this context, we are developing a coordination functionality and we are addressing work flow management support. Due to the commonalities in the underlying system architecture, users can work with SEPIA and DOLPHIN in parallel and exchange objects and structures between them without any problem. This is needed and helpful in further extensions we are currently working on (Streitz, 1994 b). This concerns the coupling of several meeting rooms in different locations and external partners in remote offices by using recent advances in network technology. This project is called “Virtual Meetings based on ATM”. While these new developments yielded significant progress, there is still a need for the evaluation of user acceptance and of scaling up in terms of the number of users and size of documents.

Acknowledgements

We like to thank B. Bokowski, L.Kirchner, J. Schümmer, and Ch. Schuckmann helping us to implement SEPIA and DOLPHIN.

References

Böhm, K. & Aberer, K. (1994). Storing HyTime Documents in an Object–Oriented Database. In Adam, N., Bhargava, B. & Yesha, Y. (eds.), *Proceedings of the Third International Conference on Information and Knowledge Management.* (pp. 26–33). ACM Press.

Conklin, J. (1987). Hypertext: An introduction and survey. *IEEE Computer*, *20*(9), 17 - 41.

Conklin, J. & Begeman, M. L. (1988). gIBIS: A Hypertext Tool for Exploratory Policy Discussion. In *Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW'88)* (pp. 140 – 152).

DeRose, S. J. & Durand, D.G. (1994). *Making Hypermedia Work. A User's Guide to HyTime.* Amsterdam: Kluwer.

Dewan, P. & Choudhary, R. (1991). Flexible user interface coupling in a collaborative system. In *Proceedings of the CHI'91* (pp. 41 – 48).

Ellis, C. A. & Gibbs, S. J. (1989). Concurrency Control in Groupware Systems. In *Proceedings of the ACM SIGMOD'89 Conference on the Management of Data* (pp. 399 – 407). ACM Press.

Ellis, C. A., Gibbs, S.J. rein, G.L. (1991). Groupware – Some Issues and Experiences. *Communications of the ACM* (January 1991), *34* (1), 38 – 58.

Fischer, D. H. & Rostek, L. (1995). *SFK: A Smalltalk Frame Kit – Concepts and Use.* Technical Report GMD-IPSI. Darmstadt, March 1995.

Garzotto, F., Paolini, P. & Schwabe, D. (1991). HDM – A Model for the Design of Hypertext Applications. In *Proceedings of the 3rd ACM Conference on Hypertext (Hypertext '91)* (pp. 313 –328).

Garzotto, F., Paolini, P. & Schwabe, D. (1993). HDM — A Model Based Approach to Hypermedia Application Design. *ACM Transactions on Information Systems, 11*, 1, 1–26.

Haake, A. (1992). CoVer: A Contextual Version Server for Hypertext Applications. In *Proceedings of the European Conference on Hypertext (ECHT'92).* (pp. 43–52) ACM Press.

Haake, A. (1994). Under CoVer: The Implementation of a Contextual Version Server for Hypertext Applications. In *Proceedings of the European Conference on Hypermedia Technology (ECHT'94)*. (pp. 81–93) ACM Press.

Haake, A. & Haake, J. (1993). Take CoVer: Exploiting Version Support in Cooperative Systems. In *Proceedings of ACM/IFIP Conference InterCHI'93.* (pp. 406 – 413). ACM Press.

Haake, J.,Bapat, A. & Knopik, T.(1994). Using a Hypermedia System for Systems Engineering. *Proceedings of the East–West International Conference on Multimedia, Hypertext, and Virtual Reality (MHVR'94)*. (pp. 63–68).

Haake, J., Neuwirth, C., Streitz, N. (1994). Coexistence and transformation of informal and formal structures: Requirements for more flexible hypermedia systems. *Proceedings of ACM European Conference on Hypermedia Technology (ECHT'94).* (pp. 1–12) ACM Press.

Haake, J. & Wilson, B. (1992). Supporting Collaborative Writing of Hyperdocuments in SEPIA. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'92)* (pp. 138–146). ACM Press.

Halasz, F. G. (1988). Reflections on NoteCards: Seven Issues for the Next Generation of Hypermedia Systems. *Communications of the ACM* (July 1988), *34*1(7), 836 – 852.

Halasz, F. G. (1991). *"Seven issues": Revisited.* Final Keynote Talk at the 3rd ACM Conference on Hypertext, San Antonio, TX, December 15 –18, 1991.

Hannemann, J., Thüring, M. & Streitz, N. A. (1990). Schreiben als Designproblem: Ein integrativer Ansatz. In D. Frey (Hrsg.), *Bericht über den 37.Kongreß der Deutschen Gesellschaft für Psychologie*. Göttingen: Hogrefe.

Hannemann, J., Thüring, M. & Friedrich, N. (1992). Hyperdocuments as user interfaces: Exploring a browsing semantic for coherent hyperdocuments. In R. Cordes & N.Streitz (eds.), *Hypertext und Hypermedia 1992* (S. 87–102). Heidelberg: Springer.

Hoschka, P., Butscher, B. Streitz, N. (1992). Telecooperation and telepresence: Technical challenges of a government distributed between Bonn and Berlin. *Informatization and the Public Sector, 2* (4) 269 – 299.

Irish, P. M. & Trigg, R. H. (1989). Supporting Collaboration in Hypermedia: Issues and Experiences. In *Journal of the American Society for Information Science*, 40(3), 192 – 199, May, 1989.

ISO (1986). ISO/IEC International Standard 8879-1986: Information Processing — Text and Office

Systems — Standard Generalized Markup Language (SGML), 1986.

Joyce, M. (1991). StorySpace as a hypertext system for writers and readers of varying ability. In *Proceedings of the 3rd ACM Conference on Hypertext* (*Hypertext '91*) (pp. 381 – 387). ACM Press.

Klas, W., Neuhold, E. J. & Schrefl, M. (1990). Using an Object-Oriented Approach to Model Multimedia Data. *Computer Communications* 13(4), 204 – 216

Knopik, T. & Bapat, A. (1994). The Role of Node and Link Types in Open Hypermedia Systems. *Proceedings of the ECHT'94 Workshop on Open Hypermedia Systems.* Report R–94–2038, Aalborg University, Denmark, October 1994, p. 33–36.

Kunz, W. & Rittel, H. (1970). *Issues as elements of information systems* (Working paper 131). Berkeley, CA: University of California.

Lakoumentas, N. (1995). *Linearisierung von Hyperdokumenten im Hypermediasystem SEPIA.* Diplomarbeit (Master Thesis). GMD–IPSI and Technical University Darmstadt. March 1995.

Lemke, A., Streitz, N. A. & Wilson, B. (1992) *WSCRAWL: The use of a shared workspace in a desktop conferencing system.* GMD-IPSI, Technical Report. Arbeitspapiere der GMD Nr. 672.

Mark, G., Haake, J., Streitz, N. (1995). The use of hypermedia in group problem solving: An evaluation of the DOLPHIN electronic meeting room environment. In *Proceedings of European Conference on Computer–Supported Cooperative Work (E–CSCW'95)* (in press).

Marshall C. (1989). Representing the structure of a legal argument. *Proceedings of 2nd International Conference on AI and Law* (pp.121–127) Vancouver.

Marshall C., Halasz, F., Rogers, R. & Jansen, W. Jr. (1991). Aquanet: A Hypertext Tool to Hold Your Knowledge in Place. In *Proceedings of the 3rd ACM Conference on Hypertext (Hypertext '91)*, (pp.261 – 275) ACM Press.

McCall, R., Bennet, P., D'Ornozio, P., Ostwald, J., Shipman, F. & Wallace, N. (1990). PHIDIAS: Integrating CAD Graphics into Dynamic Hypertext. In A. Rizk, N. Streitz, & J. André (eds.), *Hypertext: Concepts, Systems, and Applications (ECHT'90).* (pp.152 – 165) Cambridge University Press.

Nanard, J. & Nanard, M. (1991). Using Structured Types to Incorporate Knowledge in Hypertext. In *Proceedings of the 3rd ACM Conference on Hypertext (Hypertext '91)* (pp. 329 – 343) ACM Press.

Neuwirth, C. M.,Kaufer, D. S., Chandhok, R. & Morris, J. H. (1990). Issues in the Design of Computer Support for Co–authoring and Commenting. In *Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW '90)* (pp. 183 – 195) ACM Press.

Neuwirth, C. M.,Kaufer, D. S., Chandhok, R. & Morris, J. H. (1990). Issues in the Design of Computer Support for Co–authoring and Commenting. In *Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW '90)* (pp. 183 – 195) ACM Press.

Neuwirth, C. M., Chandhok, R., Kaufer, D., Erion, P., Morris, J. & Miller, D. (1992). Flexible Diff–ing in a Collaborative Writing System. In *Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW '92)* (pp. 147 – 162) ACM Press.

Newell, A. (1980). Reasoning, problem solving, and decision processes: The problem space as a fundamental category. In R.S. Nickerson (ed.), *Attention and performance* (Vol. 8, pp. 693 - 718). Hillsdale, NJ.: Lawrence Erlbaum.

Newcomb, S. R., Kipp, N. & Newcomb, V. (1991). The ''HyTime" Hypermedia / Time–based Document Structuring Language. *Communications of the ACM*, *34* (11).

Rada, R., Wang, W., & Birchall, A. (1992). An Expertext System for Collaborative Authoring. *Expert Systems with Applications*, *5*, 3/4, 275–288.

Rada, R., Wang, W., & Birchall, A.(1993). Retrieval Hierarchies in Hypertext. *Journal of Information Processing and Management*, *29*, 2, 359–371.

Rein, G. L. & Ellis, C. A. (1991). rIBIS: a real-time group hypertext system. *International Journal of Man-Machine Studies*, *34*, 339 – 367.

Schütt, H. & Haake, J. M. (1993). Server Support for Cooperative Hypermedia Environments. In H. P. Frei & P. Schäuble (eds.), *Proceedings of Hypermedia'93* (pp. 45–56). Heidelberg: Springer.

Schütt, H. & Streitz, N. A. (1990). *HyperBase*: A Hypermedia Engine Based on a Relational Database Management System. In A. Rizk, N. Streitz, & J. André (eds.), *Hypertext: Concepts, Systems, and Applications (ECHT'90)* (pp. 95–108). Cambridge: Cambridge University Press.

Schuler, W., Hannemann, J., Streitz, N. (eds.) (1995) *Designing User Interfaces for Hypermedia.* Heidelberg: Springer.

Schuler, W. & Smith, J.B. (1990). Author's Argumentation Assistant (AAA): A Hypertext–Based Authoring Tool for Argumentative Texts. In A. Rizk,

N. Streitz, & J. André (eds.), *Hypertext: Concepts, Systems, and Applications (ECHT'90)* (pp. 95–108). Cambridge: Cambridge University Press.

Smith, J. B., Weiss, S. F. & Ferguson, G. J. (1987). A Hypertext Writing Environment and its Cognitive Basis. In *Proceedings of the First ACM Workshop on Hypertext (Hypertext'87)*(pp.195 – 214) ACM.

Smith, J. B. & Smith, F. D. (1991). ABC: A hypermedia system for artifact–based collaboration. In *Proceedings of the 3rd ACM Conference on Hypertext* (*Hypertext '91*) (pp. 179 – 192). ACM Press.

Smith Catlin, K., Garrett, L. N. & Launhardt, J. A. (1991). Hypermedia Templates: An Author's Tool. In *Proceedings of the 3rd ACM Conference on Hypertext (Hypertext '91)* (pp. 147 – 160) ACM Press

Smolensky, P., Bell, B., Fox, B., King, R. & Lewis, C. (1987). Constraint-Based Hypertext for Argumentation. In *Proceedings of the First ACM Workshop on Hypertext (Hypertext'87)*(pp. 215 – 245).

Streitz, N. (1987). Cognitive compatibility as a central issue in human-computer interaction: Theoretical framework and empirical findings. In G. Salvendy (Ed.), *Cognitive engineering in the design of human-computer interaction and expert systems* (pp. 75-82). Amsterdam: Elsevier.

Streitz, N. (1994 a). Putting objects to work: Hypermedia as the subject matter and the medium for computer–supported cooperative work. In M. Tokoro & R. Pareschi (eds.), *Object–Oriented Programming. Proceedings of ECOOP'94.* LNCS 821 (pp. 183–193). Heidelberg: Springer.

Streitz, N. (1994 b). From Individual Work and Desktop–based Collaboration to Virtual Meeting Room Environments. Keynote talk at the *East–West International Conference on Multimedia, Hypertext, and Virtual Reality (MHVR'94)* Moscow. September 14–16, 1994.

Streitz, N. (1995). Designing Hypermedia: A Collaborative Activity. *Communications of the ACM* (August 1995) *38* (8), 70–71.

Streitz, N., Geissler, J. Haake, J., Hol, J. (1994). DOLPHIN: Integrated meeting support across LiveBoards, local and remote desktop environments. In *Proceedings of the ACM Conference on Computer–Supported Cooperative Work (CSCW'94)* (pp. 345 – 358). ACM Press.

Streitz, N.A., Hannemann, J. & Thüring, M. (1989). From Ideas and Arguments to Hyperdocuments: Travelling through Activity Spaces. In *Proceedings of the 2nd ACM Conference on Hypertext (Hypertext'89)* (pp. 343 – 364). ACM Press.

Tatar, D. G., Foster, G. & Bobrow, D. (1991). Design for conversation: lessons from Cognoter. *International Journal of Man-Machine Studies*, *34*, 185–209.

Thüring, M. , Haake, J. & Hannemann, J. (1991). What's Eliza doing in the Chinese Room? - Incoherent hyperdocuments - and how to avoid them. In *Proceedings of the 3rd ACM Conference on Hypertext* (*Hypertext '91*) (pp. 161 – 177). ACM Press.

Thüring, M., Hannemann, J., Haake, J. (1995). Hypermedia and Cognition: Designing for Comprehension. *Communications of the ACM* (August 1995) *38* (8), 57–66.

Toulmin, S. (1959). *The uses of argument*. Cambridge: Cambridge University Press.

Wäsch, J. & Aberer, K. (1995). Flexible Design and Efficient Implementation of a Hypermedia Document Database System by Tailoring Semantic Relationships. In *Proceedings of the Sixth IFIP Conference on Data Semantics (DS–6).*

Walker, J. (1991). Netfone Release 1. Posted in newsgroup *comp.sources.sun*. September 12, 1991.

Wang, W. & Rada, R. (1995). Experiences with Semantic Net–based Hypermedia. *International Journal of Human-Computer Studies* (in press).

Wilson, B. (1992). WScrawl 2.0. Available at http://www–cs–students.stanford.edu/~brianw/bw.html

You, G.–N. & Rada, R. (1994). A Systematic Approach to Outline Manipulation. *International Journal of Human-Computer Studies 41,* 283–308.