

Collaborative Authoring of Hypermedia Documents

Jörg M. Haake* & Christine M. Neuwirth**

* Integrated Publication and Information Systems Institute (IPSI), Gesellschaft für Mathematik und Datenverarbeitung (GMD), Dolivostr. 15, D-64293 Darmstadt, haake@darmstadt.gmd.de

**Carnegie Mellon University, Pittsburgh, PA 15213 USA, cmn@cmu.edu

This paper identifies issues in computer-support for collaborative authoring of hyperdocuments and shows how SEPIA, a cooperative authoring environment, addresses these issues. First, hyperdocuments can be used to create and maintain technical documentation. Second, activity spaces support the cognitive and social processes involved in the creation of hyperdocuments. Third, a shared hyperdocument database, versioning, different modes of collaborative work, and awareness of the activities of other group members support asynchronous and synchronous distributed collaboration, as well as smooth transitions between them. Fourth, annotations can be used to communicate about drafts and plans. Although initial experience with SEPIA indicates that it provides strong dedicated support for collaborative writing of hyperdocuments, we identify annotations as one area where further improvement is possible and outline issues involved in providing better support for generating, receiving, and reacting to annotations.

Keywords: hyperdocuments, collaborative authoring support, SEPIA, annotations

INTRODUCTION

Technical writers are making increased use of hypermedia documents to facilitate the maintenance and even the original authoring of technical documents (Glushko, 11; Walker, 42). In a hypermedia document, for example, a single piece of information can be stored in only one location, even though it may be used in many places in a document or set of documents. This is accomplished through what is called a hypermedia architecture: The document contents are segmented into units called "nodes" and the document is put together by linking the nodes together (so any given node can be linked into more than one place). The structure of nodes and links that results often resembles a network, though a reader may have the impression of reading a (traditional) linear document when following a predefined "path" of nodes and links that a technical writer has provided.¹ A hypermedia architecture can facilitate the maintenance of information in technical documents: Since the information is only stored once even though it may be displayed in many locations, a single revision of the information will change the content everywhere it appears. As we shall illustrate in this paper, a hypermedia architecture can facilitate the collaborative authoring of technical documents as well, by providing the underlying architectural support needed to support many collaborative writing activities.

Surveys of writers in business and industry indicate that the majority of documents, especially technical documents, are written collaboratively (Ede & Lunsford, 7). Recent advances in computer

¹Whereas a traditional document has only one (linear) dimension, a hypermedia document, which may consist of a network of nodes and links, may have many dimensions. Drawing on the use of word "hyper" in mathematics to refer to spaces of n-ary dimensions, Ted Nelson (1967) coined the word "hypertext" to express this multidimensional property of such documents. The term "hypertext," while still in widespread use, evolved to "hypermedia" to emphasize the multimedia contents of current electronic documents (i.e., not only text but also graphics, video, sound, animations, etc.).

and network technologies have brought about increasing interest in providing computer support for such collaborative writing practices. Unlike traditional word processors, which at best only support the cognitive activities of writers, technologies for supporting collaborative writing must also provide support for writers' social needs and for their co-ordination needs. Furthermore, recent advances in network and distributed systems technology have brought about increasing interest in providing computer support for collaborative work at a distance. In many cases, a group of distributed co-workers (such as technical authors) work collaboratively on a common task (such as the production of a technical document).

This paper outlines issues in computer support for collaborative authoring, reviews collaborative authoring support, and discusses how SEPIA, a research prototype for collaborative authoring of hypermedia documents, meets these requirements. One major possibility for improving SEPIA concerns annotation support. The paper elaborates the issues surrounding such support. Finally, the paper outlines directions for future research with the SEPIA prototype.

ISSUES IN COMPUTER-SUPPORTED COLLABORATIVE AUTHORING

In identifying issues for computer support for any collaborative task, it is important to consider three sources:

- 1) the task itself (e.g., in this paper, the task of writing),
- 2) the process of collaboratively performing the task (e.g., collaborative writing), and
- 3) the need to communicate when collaboratively performing the task (e.g., communicating when writing collaboratively).

For collaborative authoring, these three aspects require an analysis of the collaborative writing process. From this analysis, we identify issues that must be addressed by systems trying to support collaborative writing.

Issues in writing

As noted in the introduction, technical writers are making increased use of hypermedia documents to support such activities as the assembly and maintenance of documentation. Compared to traditional linear documents, hypermedia documents allow for new kinds of organizations (e.g., the construction of information networks). For example, writers may define a network of concepts linked to definitions. To accomplish this, a set of building blocks consisting of "concept" and "definition" nodes together with "definition" links is useful. To support the design of such new kind of hyperdocuments, tools such as building blocks, means for revising building blocks and their connections, and guidelines for using building blocks effectively to design hyperdocuments are needed. Guidelines can take the form of "templates," and writers, especially technical writers, often employ them to ensure consistency and to maintain a predefined organization and format for a document. In our example, a "definition" template ensuring that every "concept" is linked to its "definition" would help to maintain consistency in the hyperdocument. Even with templates, however, writers sometimes need to adapt the structure to another audience or to change the structure of the document over time. Therefore, templates as well as the restructuring of templates and documents based on them should be supported. For hypermedia documents, the definition and functionality of such templates is still an active research area.

Although the production of paper documents remains widespread, in electronic documents authors can use new multimedia contents (e.g., video, voice) that should be supported in an authoring environment.

In an analysis of requirements for computer support for the cognitive processes of individual writing, Streitz et al. (37) identify three basic activities for which writers need support: planning the document's contents (as well as other planning related to a document's production such as audience, purpose and procedure); generating and structuring domain knowledge (or content); and formulating and organizing the document from a rhetorical perspective (generating and adapting content to purpose and audience).

In each of these activities, writers often consider several alternatives before tentatively committing to any single one. For example, even though most experienced writers have some sort of plan for a document before they begin drafting, a very open-ended design task like writing sometimes requires writers to modify or even abandon those plans as they discover additional aspects of the problem during the process of drafting itself. Since authors often explore different plans, contents, and formulations for a document, the construction of alternative structures and alternative parts of a document should be supported.

Although writers often do some upfront planning of a document, the activities of planning, generating content, organizing, and so forth recur throughout most of the writing process (Hayes & Flower, 19). Thus, computer support for these activities must allow authors to engage in them in any sequence, with smooth transitions between them.

Issues in group work

The majority of technical documents are written by groups of authors that may be either co-located or distributed. Thus, computer support for the collaborative writing process must support co-located and distributed authoring of a shared electronic document.

Interviews of successful collaborative writers indicate that they engage in both synchronous and asynchronous collaboration (Kraut et al., 24), thus, both should be supported. Synchronous collaboration refers to a group of authors working on the same part of a shared document at the same time. Asynchronous collaboration refers to a single author working on a part of a document that is shared with other authors. According to Kraut et al., authors often work synchronously to plan a document and revise it, and asynchronously to draft it. For synchronous collaboration (e.g., meetings or joint editing sessions), synchronized access to the same document parts and shared views on those documents must be supported. For asynchronous collaboration (e.g., work on different document parts or work on the same document at different times) either access to the same shared document or parallel drafts with subsequent merging of alternative versions ("draft passing") should be supported. Since different collaborative activities not only require different collaboration support, but also they may change over time as authors start and finish portions of their work, smooth transitions between synchronous and asynchronous collaboration should be supported.

Coordination plays an important role in group work. It includes activities such as partitioning tasks into subtasks, dealing with dependencies between tasks, and assigning people to tasks. In very large documents, the partitioning of work may involve tasks such as technical accuracy checks, editing, layout, and so forth. Such tasks may have to be accomplished in particular sequences and by particular deadlines. Collaborative authoring systems should support such coordination processes. Depending on the context, this support could require interoperability with specialized systems used to support particular subtasks.

In addition to pre-planned coordination (e.g., task assignments), systems need to support spontaneous coordination brought about by dynamic writing activities. For example, although two authors may plan that they should work in different sections of the document, the changes an author makes in one section may have repercussions in another, so it may be desirable for an author to depart from the plan and do some work in the same section as another co-author. Co-authors who are working on the same document section may want to synchronize their changes, solve conflicts or exploit synergy effects. To detect those situations requires awareness of activities of co-workers which needs to be communicated.

Issues in communication about writing the document

Coordination reduces the amount of communication required to collaborate. On the other hand, coordination requires communication. Two different kinds of communication about writing the document can be distinguished: synchronous and asynchronous communication.

Synchronous communication about group work requires face-to-face co-location or the presence of synchronous communication channels such as audio or video links. Such communication links need to operate synergistically with the presence of synchronous collaborative authoring (e.g., joint editing, meetings).

Supporting spontaneous coordination may require communicating synchronously about (1) who is currently working (2) what they are working on and (3) how they would like to collaborate. This has been referred to as "group awareness" (Dourish & Bellotti, 6).

Asynchronous communication about group work can either be coupled to the document (e.g., annotations) or be independent from the document (e.g., separate messages). Thus, annotations as well as messaging needs to be supported. Both can be used as a means for coordinating the group work process.

REVIEW OF COLLABORATIVE AUTHORING SUPPORT

In the research literature, one can distinguish between systems aimed at supporting the collaborative authoring of documents (linear or hyperdocuments) and systems aimed at supporting conferences (the conferences might be to discuss the authoring of documents, but not necessarily). The first group we call co-authoring systems, while the second is called conferencing systems. For both groups, asynchronous and synchronous systems can be distinguished.

Coauthoring systems

A number of research efforts to provide computer support for coauthoring have focused on supporting asynchronous coauthoring, including many hypermedia authoring systems as well as shared editing systems for linear documents. Asynchronous hypermedia authoring systems either follow a draft passing approach (and may in addition log changes made by different authors) like in NoteCards (Halasz, 16; Irish & Trigg, 20), or they allow asynchronous access to shared nodes (i.e., two or more authors can access the same node but at different times) like in KMS (Akscyn et al., 1) or CoAuthor² (Hahn et al., 18). InterNote provided asynchronous access to shared nodes and also supported asynchronous communication by means of annotations (Catlin et al., 4). Similarly, asynchronous shared editing systems for linear documents support draft passing and elaborated annotation facilities like in the PREP Editor system (Neuwirth et al., 29) or they allow asynchronous work on a shared document like in Quilt (Fish et al., 10).

Likewise, a number of research efforts have focused on supporting synchronous writing. Synchronous shared editing systems provide authors with strict shared views (WYSIWIS - "What You See (on your screen) Is What I See (on my screen)") or relaxed shared views (what you see on the screen may be what I see, but not necessarily) (Stefik et al., 36) as well as concurrent shared editing capabilities (authors can edit the same part of the document at the same time). According to the media supported (text vs. graphics), one can distinguish shared text editors like GROVE (Ellis et al., 8) and ShrEdit (McGuffin & Olson, 27) that provide shared views, concurrent editing (GROVE) or locked work areas (ShrEdit) and presentation of group status information. Shared graphics editors like CaveDraw (Lu & Mantei, 26), GroupSketch (Greenberg & Bohnet, 12), TeamPaint (Ishii et al., 22), and

² In addition, CoAuthor provides a real-time conferencing system, supporting synchronous discussions among authors.

WSCRAWL (Wilson, 44) provide concurrent editing, layers (CaveDraw, TeamPaint), and telepointers.³ In addition, TeamPaint provides integrated audio and video communication.

There are only a few coauthoring systems supporting both asynchronous and synchronous collaborative authoring in one integrated system. rIBIS (Rein & Ellis, 32) is a distributed group hypertext system allowing a group of authors to simultaneously read and edit a hypertext. To support asynchronous and synchronous authoring, it provides two modes of interaction which authors must explicitly select: The "Loosely-coupled mode" allows multiple users to access data at the same time. rIBIS synchronizes changes made by different users. These changes are not automatically reflected on other users' interfaces; The "tightly-coupled mode" provides shared views and a turn-taking floor control policy.⁴ There is only one tightly-coupled session per hypertext. Other systems integrate asynchronous and synchronous authoring for linear documents. SASSE (Baecker et al., 2) is a shared editing system for supporting different activities of coauthoring. It provides collaboration awareness and annotation facilities. Aspects (Group Technologies, 13) provides explicit sessions with different floor control policies. Within a session, shared editors for text and graphics can be opened to support joint editing.

Current coauthoring systems only partially address the issues in computer-supported collaborative authoring. In particular, no single system addresses all of the following issues:

- ◆ with respect to writing: planning, multimedia contents, and design of hypermedia documents,
- ◆ with respect to group writing: smooth transitions between collaborative modes, coordination,
- ◆ with respect to communication about group writing: tightly integrated support for synchronous communication channels.

Conferencing systems

Groups of authors and reviewers, especially groups whose members are distributed, sometimes use conferencing systems to communicate about writing.

Asynchronous conference systems include e-mail and "bulletin boards." Bulletin boards allow groups to define electronic spaces for discussions on particular topics. For example, the USENET system (Todinao, 40) allows multiple discussions (called newsgroups), and supports both text and graphics contents.

Synchronous conferencing systems include desktop teleconferencing systems and meeting support systems. Desktop teleconferencing denotes conferences that are held using computers rather than meeting face-to-face. Each user has a workstation that is networked with the other users' workstations. With a desktop teleconferencing program all users may share windows on their workstations' displays and may use digital communication links (such as audio channels). Desktop teleconferencing systems like MMConf (Crowley et al., 5) manage distributed conferences, the screen space within a conference and provide a framework for building multimedia group applications. The MERMAID desktop conferencing system (Watanabe et al., 43) integrates voice and video communication with shared multimedia documents and conference management. Other systems like ClearFace (Ishii & Arita, 21) or TeamPaint (Ishii et al., 22) integrate the construction of an artifact (e.g., a drawing) with audio and video communications between two users. In contrast, meeting support systems usually support meeting rooms and are not aimed at document production (for more details see Nunamaker et al., 31).

³ A telepointer is usually displayed as an arrow with the name of its user. It is simultaneously visible on each user's screen and its location is coupled to the cursor position of the user's mouse. Thus it can be used for pointing and gesturing.

⁴ The term "floor control policy" refers to the way potential conflicts over shared resources are handled. For example, a turn taking policy is a policy in which different users ask for the shared resource with each user getting a turn.

Current conferencing systems only partially address the issues in computer-supported collaborative authoring. In particular, no single system addresses all of the following issues:

- ◆ with respect to writing: planning, production of (hypermedia) documents
- ◆ with respect to group writing: integration of asynchronous and synchronous writing, coordination,
- ◆ with respect to communication about group writing: the notion of a document.

In the next three sections, we show how the issues in computer-supported collaborative authoring are addressed in a single, integrated cooperative authoring system, SEPIA. SEPIA (Structured Elicitation and Processing of Ideas for Authoring) is a cooperative hypermedia authoring environment which enables a distributed group of authors to work collaboratively on a shared hyperdocument, both asynchronously and synchronously (Streitz et al., 38). In particular, in the next three sections, we will discuss how SEPIA supports

- ◆ writing,
- ◆ group work, and
- ◆ communication about documents.

SUPPORT FOR WRITING IN SEPIA

The **product** of the authoring activity in SEPIA is a hypermedia document. For example, Figure 1 depicts a hypermedia document, "System Manuals," consisting of six subsections: "Introduction", "Overview," "Tutorial," "Installation & Maintenance," "Administrator's Guide" and "Problem Detection & Solving." The node labelled "Introduction" is opened in a new text editor window (entitled "Sepia Text Editor") to show its content ("This hyperdocument presents a set of manuals...") and the node labelled "Overview" is likewise opened to show its content, which is not a text, but a substructure. From a technical perspective, the hyperdocument consists of multimedia atomic nodes, composite nodes and links. Atomic nodes (such as "Introduction" in Figure 1) contain a set of content objects, each representing a chunk of information (e.g., "This hyperdocument presents...") of a different media type (e.g., text, graphics, audio, video). Composite nodes contain other nodes and links (e.g., the composite "Overview" contains five other nodes: "Introduction to Overview", "Purpose of the System", "Methodology of Use", "Main Concepts" and "Examples of Use"), and these can be nested (e.g., the composite node, "Methodology of Use" is nested inside the composite node "Overview"). Nodes and links are named (i.e., nodes are represented as boxes carrying the node name and its type as a label: links are represented as arrows carrying the link type as a label).

Special types of composite nodes and links can be used to construct different types of paths (Zellwegger, 45) such as

- (1) linear paths. For example, the node labeled "Overview" in Figure 1 is a linear path node that presents readers with a linear sequence of subnodes, "Introduction to Overview," "Purpose." and "Methodology of Use."
- (2) alternative paths. For example, the document labeled "System Manuals" in Figure 1 presents readers with four alternative paths that readers can choose to follow: "Overview," "Installation & Maintenance," "Administrator's Guide," and "Problem Detection & Solving."
- (3) conditional paths (i.e., paths that present readers with a choice of which path to follow). Although none are depicted in Figure 1, an example would be a path that allowed a reader to decide whether to read a section of a tutorial, based upon whether the reader is already familiar or not with some basic prerequisite concepts, such as word processing.

- (4) non-linear networks (i.e., composites that allow readers to explore a network of nodes and links). For example, the node labeled "Main Concepts" in Figure 1 contains a non-linear network. When opened (see bottom, right window, Figure 1), it allows readers to browse among definitions of all the concepts in the system (e.g., the concept that a document consists of entities is expressed by the node-link-node structure "Document" → "consists of" → "Entity").

Note that Figure 1 shows the author's view of the hyperdocument. For authors of hyperdocuments, contents and structural relationships between the constituents are of primary interest. Therefore, the structure of the hyperdocument is presented graphically in SEPIA. In contrast, readers can use a completely different reader program for the presentation of and interaction with the hyperdocument. Such a dedicated reader interface has been described as SEPIA's Presentation Interface (Hannemann et al., 17). To overcome the separation between authoring and reading environment, a unified document model or a translator between the different document formats is required.

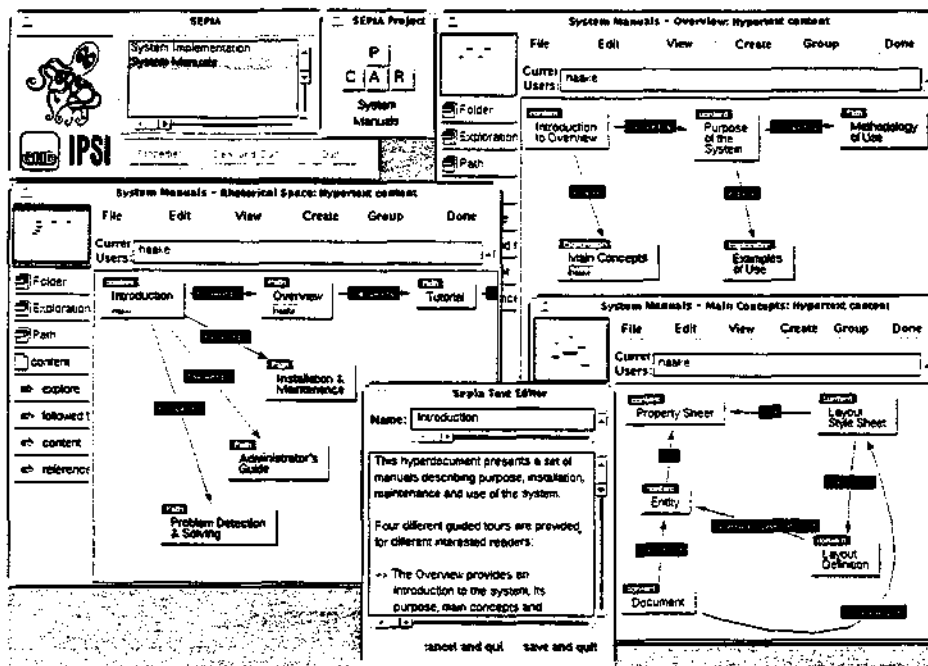


Figure 1: Author's view of a hyperdocument under construction in SEPIA's rhetorical space (see below)

The **process** of authoring is supported by the concept of activity spaces. An activity space provides problem-specific objects and operations to facilitate the author's activities when working on the problem. For writing, three subproblems are supported by specific activity spaces:

- ◆ the planning space for planning the document's content and construction process,
- ◆ the content space for idea dumping, information gathering, and structuring of background information, and
- ◆ the rhetorical space for creating the final hyperdocument for a specific target audience.

Since argumentation plays an important role in writing for a large number of document types, the three spaces are supplemented by a fourth space:

- ◆ the argumentation space for representing lines of argumentation.

These four activity spaces present typed⁵ hypermedia objects for solving the related subproblems of writing. By providing dedicated hypermedia object types for each subproblem, the basic activities of writing are supported. Each activity space is realized as a graphical browser. Figure 2 shows a screendump of the SEPIA activity spaces.

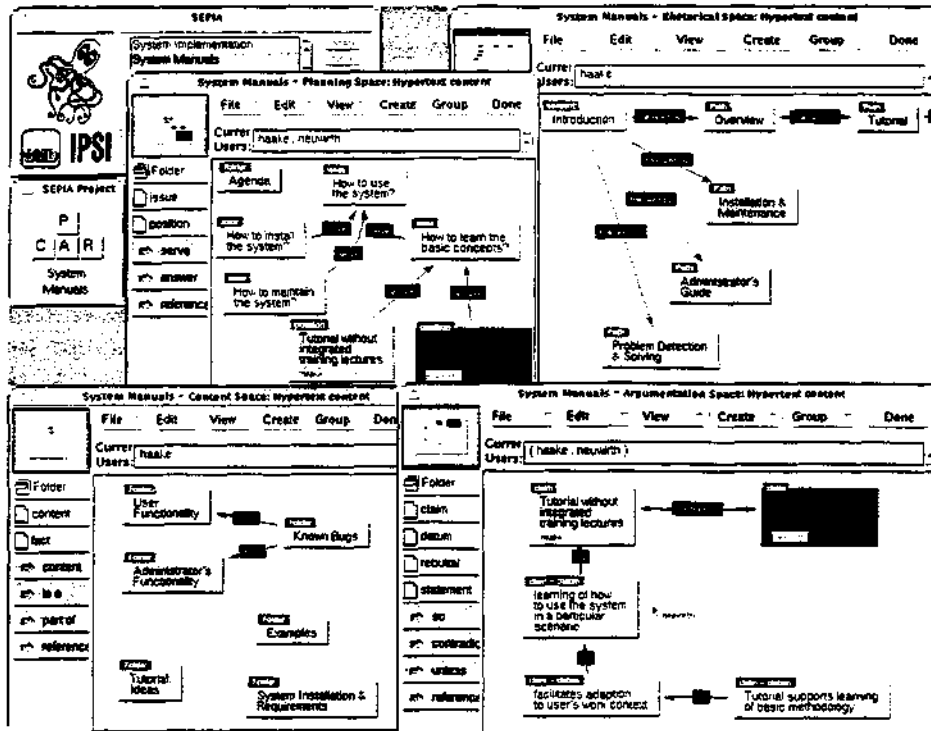


Figure 2: User-interface of SEPIA

In the planning space, authors have the opportunity to externalize their writing plans, to construct issues to be concerned with in the document, and to establish an agenda for the authoring activity. For example, the Planning Space node in Figure 2 (top, left) depicts (1) a node, "How to use the System," that represents the top-level node in a set of author-constructed issues, subissues (e.g., "How to Learn Concepts") and positions on issues (e.g., "Tutorials without training lectures" vs. "Tutorials with training lectures"); and (2) the composite node "Agenda" that contains plans for how to proceed with writing. This space serves as a meta-space for coordinating the activities in the other three spaces and for controlling the progress of the writing process.

The formulation of issues may lead to

- ◆ the identification of important topics which are elaborated in the content space,
- ◆ the generation of positions and claims which have to be supported in the argumentation space,
- ◆ and the creation of the document in the rhetorical space.

For the development of an issue structure, the author can rely on a set of dedicated nodes and links. Here, we use a modification of the IBIS method (Kunz & Rittel, 25), which involves explicitly representing issues and positions on issues. In addition, the planning space is linked to the argumentation space. Positions which are formulated as answers to issues in the planning space are also displayed as claims in the argumentation space, prompting the author for providing pro and con arguments.

The objects and operations of the content space are dedicated to facilitate the development of a domain model. Figure 2 depicts a Content Space (bottom, left) containing content related to the system that is being documented, such as "User Functionality," "Known Bugs," and so forth. This

⁵ "Types" are used in SEPIA to constrain the use of hypermedia objects, such as constraining a composite of type "Glossary" to include only hypermedia nodes of type "Term Definition". The possible sources and destinations of links can be constrained as well. Thus, types support the construction of well-formed hyperdocuments.

space can also allow users to access background material either from internal (e.g., previous documents) or external sources (e.g., querying a data base). For this purpose, SEPIA uses the structuring facility of hypertext to support dumping and collecting ideas in nodes with textual and graphical content, grouping them in topic-related clusters by composite nodes, and connecting them via typed links.

In the rhetorical space, the author creates a reader-oriented document. Figure 2 depicts the Rhetorical Space (top, right), which was also depicted in Figure 1. To a large degree but not exclusively, it is produced on the basis of transformations of the material in the content space. This final product can be a conventional, linear text or a hyperdocument consisting of a network of nodes and links. Both document types constitute a scale extending from strictly linear to strictly non-linear documents. Since hyperdocuments can vary in the degree of their linearity, they can be very different with respect to their structure and presentation. 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 construction kit for designing coherent hyperdocuments (Thüring et al., 39).

The Argumentation Space supports the development of an argumentative structure by providing appropriate design objects and operations based on an extension of the well known argumentation schema developed by Toulmin (41). Using the argumentation space, the author can elaborate an argumentation by generating support or objections on different levels, by formulating contradictions and by constructing argumentative chains (for details see Streitz et al., 37). Figure 2 (bottom, right) depicts an Argumentation space in which the merits of producing a "Tutorial without Training Sessions" versus a "Tutorial with Training Sessions" are being discussed.

When "travelling through activity spaces," authors do not need to follow a predetermined route. At every point in the design process, authors can decide which subspace to use next. In every subspace, they can externalize intermediate results, generate new ideas and revise design decisions. To guarantee such flexibility, it is necessary to facilitate the interaction and smooth transformation of knowledge between the different activity spaces. In SEPIA, this is accomplished by the automatic transfer of design objects between specified spaces, their reuse, and the indication and control of references between activity spaces. Thus, moving between activity spaces supports the opportunistic nature of writing.

To support authors in exploring alternative content and structure, SEPIA allows authors to create different hypermedia structures in any activity space. To guide the authoring of documents with pre-defined structures, empty template documents can be copied and used as a skeleton for the document structure. This approach can be further developed to support dynamic expansion of templates as proposed by Smith Catlin & Garrett (34). Restructuring is supported by operations for removing and creating links as well as for moving objects between composite nodes.

SUPPORT FOR GROUP WORK IN SEPIA

To support a distributed group of authors who work synchronously on the same document, SEPIA provides access to a shared hyperdocument database⁶ that supports for asynchronous and synchronous collaboration.

- ◆ For asynchronous collaboration, the integration of a version server allows SEPIA to support both separate work of several users on different versions of the same document part

⁶ SEPIA exploits a client-server architecture, where the server provides services to a number of clients and each clients can request the server to perform some service. The advantage of the client-server approach is that the central server can synchronize requests from several clients while each client may run on a separate host, not necessarily knowing about other clients. Thus, different SEPIA clients can access the same hypermedia objects. The central database server ensures consistency between different SEPIA clients by employing locks and a cooperative transaction schema (for more details see Streitz et al., 1992; Schütt & Haake, 1993).

as well as **isolated work** of one author (Haake & Haake, 15). The latter disables other authors from changing the encapsulated workspace of the author working in isolated work mode. While separate work still enables other authors to create new alternative versions or to join the first author's work, isolated work is required to constitute an encapsulated workspace needed to integrate alternative versions into a single version.

- ◆ For synchronous collaboration, SEPIA provides two more work modes to support different degrees of collaboration (see figure 2):

In **loosely-coupled work**, several authors share the same version of a hypermedia object (i.e., each of them has an active browser on the corresponding atomic or composite node). Thus, the respective browsers are initially in loosely-coupled mode. Authors are made aware of each other via (1) a list of all concurrent authors in the respective browsers, (2) highlighting of objects selected or used by other users and displaying the names of those users as name tags of the objects, (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. Cooperative transactions are used to prevent authors from simultaneously modifying the same object properties (Schütt & Haake, 33).

In **tightly-coupled work**, several authors share the same version of a hypermedia object and the same view on that object. Thus, the respective tightly-coupled browsers display a WYSIWIS-view on the object's content. In addition to the functionality of loosely-coupled mode, scrolling and resizing events are immediately broadcast to all tightly-coupled browsers.

In summary, loosely- and tightly-coupled work mode provide relaxed and strict WYSIWIS views.

Awareness of the coauthors' activities is a prerequisite for smooth and ad hoc transitions from one mode of collaboration to another. Currently, the transition from asynchronous (individual) work mode to loosely-coupled work mode is triggered automatically when a second author opens a version of an object already viewed by the first author. This is indicated by a "door bell" sound on both workstations and the change of the user list. While in loosely-coupled mode, authors might want to begin a tightly-coupled session. To start a tightly-coupled session, one author selects all or a subset of those authors currently working on the same object and invites them to participate in the session. The system asks each of them to confirm. The browsers of those authors who confirmed are shifted into tightly-coupled mode. Authors can exit a tightly-coupled session either by closing the browser or by returning to loosely-coupled mode (for details see Haake & Wilson, 14). Note, that SEPIA supports multiple coupled sessions. Authors can be in different collaboration modes in different browsers.

Coordination is partially supported by SEPIA's planning space. It allows authors the construction of work plans and work distributions. Furthermore, tightly-coupled sessions can be used to satisfy the spontaneous need for coordination triggered by conflicting changes made by synchronous collaborators or by possibilities for closer collaboration. Currently, we plan to extend SEPIA with a fifth activity space called the coordination space. In this space, task structures managing the workflow and mechanisms for visualizing the status of the collaboration will be provided.

SUPPORT FOR COMMUNICATION ABOUT WRITING DOCUMENTS IN SEPIA

Two situations in communicating about writing documents can be distinguished: synchronous and asynchronous communication.

Synchronous communication about writing documents

Synchronous communication about group work on documents include

- ◆ Face-to-face meetings. SEPIA can be used to facilitate co-located meetings (see also future work in the summary).
- ◆ Verbal communication among tightly-coupled authors. Beyond the traditional telephone, SEPIA supports a digital audio connection between the members of a tightly-coupled session. Using a custom-build analog video network, two authors can communicate via a video link. To support online discussions, tightly-coupled browsers provide a telepointer for each member of the session. These can be used for pointing and gesturing while talking simultaneously.
- ◆ Implicit communication about concurrent activities. The group awareness provided by SEPIA is used to visualize the group status in the user list of each browser (i.e., in figure 2, the user list shows tightly-coupled sessions by enclosing the list of member names in brackets and listing all loosely-coupled users' names after the list of tightly-coupled sessions). Furthermore, the automatic propagation of modifications of objects to all loosely- and tightly-coupled browsers together with the highlighting and name-tagging enables authors to monitor the synchronous activities of their co-workers. In addition, the versioning system can be used to ask for activities performed by others in a certain time interval (for more details see Haake & Haake, 15).

Asynchronous communication about writing documents

Two primary means of communicating asynchronously about documents have evolved: (1) an "overall review" in which the reviewer writes an evaluation about the primary document that may include recommendations for changes but in which specific words and lines of the document are seldom referenced, and (2) a "document-based review" in which the reviewer usually annotates directly on a copy of the document, often referencing particular parts and suggesting specific changes. Although there are some ways in which electronic mail systems might be augmented to support the process of "overall review" of documents (e.g., a "reply-expected-by-<date>" function which would cause the mail system to send a reminder to a tardy reviewer), electronic mail seems well-suited to electronic asynchronous communication about documents (for a review of electronic mail and some of its effects on communication within organizations, see Sproull & Kiesler, 35). On the other hand, although electronic mail (or another sort of file transfer program) is one of the primary means by which authors carry out "document-based reviews" asynchronously (the other means is by sending physical hard copies of documents), it seems less well-suited to document-based review, because it lacks facilities to support annotating documents.

SEPIA supports annotating documents by specific annotation node and link types. Annotation nodes can be connected to any hypertext object via annotation links. Annotation nodes are atomic nodes that may contain content of any media (e.g., text, audio, graphics). For example, authors can attach voice comments to a piece of text. Anchoring annotation links to single atomic nodes, links or composite nodes (e.g., paths) enables authors to annotate single chunks of information, relationships or whole subgraphs of the document (e.g., a section represented by a path). Annotations are presented to authors as regular hypertext objects in SEPIA's graph browsers.

ISSUES IN SUPPORTING ANNOTATIONS

Even though SEPIA's annotation facilities enable co-authors to comment on a shared hyperdocument, our initial experience with annotations in SEPIA indicate that more sophisticated support is

needed. For example, authors don't like to see the original hyperdocument and all annotations made to it simultaneously (since the concurrent presentation requires space and thus changes the original layout).

Since annotations are one of the major ways authors communicate about documents (even authors who can meet face-to-face often annotate each other's documents asynchronously, then hold face-to-face meetings to discuss the annotations), better support for annotations in collaborative hypermedia systems remains an active research area. The next section outlines the requirements for annotations that such systems need to provide.

Generating annotations

Reviewers working with paper documents are able to create annotations easily, by putting a pen to paper and executing a set of simple and well-practiced gestures, such as circling parts of the document, drawing arrows, crossing-out parts of text, and writing or drawing in the margins and other blank spaces in a document. The ease with which reviewers can annotate paper documents should also be supported in hypermedia documents. In particular, it is important that the system minimize the number of actions actually required to generate an annotation. For example, in traditional, paper-based systems of annotation, *creating* an annotation typically requires the user to carry out one simple action to begin the task – put the pen to the page in a margin next to the place in the text the user wants to write. In many hypertext systems, the user must carry out four actions to accomplish the same task: create a node for the annotation, create a link, specify a "from" node and specify a "to" node – all before beginning to write. A system to support annotations should support generating annotations with simple actions.

Hypermedia documents, however, are far more complex than paper documents, and these differences give rise to a number of design issues in supporting the generation of annotations that need to be addressed. First, the generation of hypermedia annotations presents far more possibilities for input devices than paper annotations (e.g., mouse, keyboard and voice as well as pen); second, the set of objects to annotate contains many more types of elements (e.g., animations, audio and video as well as text and drawings); third, the structure of the documents is more complex (e.g., non-linear texts as well as linear); and finally, the views on the documents are more flexible (e.g., a graph view of the hypertext structure as well as a linearized view).

The little research that exists on annotations supports the contention that a system to support annotations should provide users with a variety of input devices. For example, reviewers reported a preference for making low-level annotations (e.g., spelling and grammar) in writing and high-level annotations (e.g., purpose, audience and content) in voice (Chalfonte, Fish, & Kraut, 4). Although users report a preference for making low-level comments in writing and high-level comments in voice, the system should not legislate the choice of input device. One can imagine, for example, a system that restricts pen input to gestures and requires the user to generate text with the keyboard, since, short of handwriting recognition, the system cannot automatically insert text that was handwritten by a pen. A system, however, should support both forms of inputting text: While an author would probably prefer keyboarded text (both because of increased readability and the possibility that the system can automatically incorporate suggested changes or they can be incorporated through a simple 'cut & paste' rather than through keyboarding the handwritten text), a reviewer with poor keyboarding skills may prefer handwritten text. Despite a preference for keyboarded text, an author may be willing to "put up with" handwritten text from a reviewer whose opinion the author values highly.

In addition to a wider variety of input devices, hypermedia documents have a wider variety of objects that might potentially be annotated. These objects require somewhat different interfaces for generating annotations. For example, a widespread technique to indicate the scope of an annotation in a text is to circle the text. In a drawing, however, a circle could be highly ambiguous and misinterpreted as part of the drawing rather than as an indicator of the annotation's scope. Different objects have different requirements for the display of annotations as well. For example, except during the last stages of text review, when the text format itself might be the subject of commentary, many text

formatting features such as the spacing between the lines are not important and can be exploited by an annotation system. To make more room for generating the text in annotations, for instance, the annotation system can expand the leading between lines of the document or between paragraphs. In a drawing, however, the relationship of one part of the drawing to another is more likely to be important, and other techniques for finding or creating space in which to display the annotation must be exploited. This discussion suggests that a system to support annotations must be able to (1) allow the user to adjust the display of annotations to various objects (e.g., text vs. drawings) and contexts of objects (e.g., early drafts of texts vs. camera-ready drafts of texts) as well as the dynamically changing size of the window in which the objects are being displayed, or (2) use a set of presentation display heuristics to automatically adjust the display. Although the incorporation of animations and video in hypermedia documents is not yet widespread, any system to support annotations should be extensible to those objects as well.

Although most hypermedia systems support some sort of linearization of hypermedia documents, the structure of a hypermedia document can be far more complex than a linear document. For example, in SEPIA's graphical activity space browsers, it is possible to reach any given node from more than one place. The question arises as to whether annotations on a node should always be displayed or should they be displayed conditionally, depending on the path by which the node was reached. Given that some annotations (e.g., concerning a transition from, for example, node1 to node2 may not be sensible if the person reading them did not reach node2 from node1), a system to support annotations should support displaying them conditional on the path.

Paper-based documents are restricted to a single view of the document. Electronic texts, however, allow multiple views (e.g., an outline view, a node-link view). A system to support annotations must support view-specific annotations. For example, suppose a reviewer indicates on an outline view that a subsection should be moved to another location and in the expanded view indicates that the subsection title should be revised. While an argument can be made that the move operation should be represented in the linear view, the annotation to revise the subsection title could be distracting and cluttering if it also appeared in the outline view.

Receiving annotations

In order to allow readers to orient themselves to the text and annotations quickly, the primary text should be easily distinguishable from the annotations. In paper documents, the text and the annotations are usually easily distinguished: the author's text is often typewritten and the reviewer's handwritten; the reviewer often chooses a colored pen when making changes on the text itself or makes marks in the margin to draw the author's attention to interlinear annotations. In computer documents, these distinctions can be made through typographic conventions, such as different type size, color, and so forth.

The system must allow readers to track the relationship between the primary text and the annotation. This includes the scope of the annotation and its location. In paper documents, often the reviewer aligns the annotation horizontally to the text to which it refers, provided there is space. This approach allows the reader to scan from primary text across to the annotation rapidly. If there is a lack of space, often a reviewer draws an arrow from the text to the annotation. In even longer annotations, the reviewer may establish a "footnote" system, with numbers in the text corresponding to numbers on the back side of the paper or on a separate sheet. Given the layout for some electronic documents is dynamic, it might be useful for the system to compute the best way of indicating the relationship between primary text and annotation. Of course, such automatic display should be under user control.

An annotation system must support a variety of strategies for reading annotations. In reading annotations, some authors scan across a set of comments, both to come to some understanding of the relationship among them and to make some overall plans for how to proceed with responses and revisions; then authors proceed with successive processing, dealing with each annotation in turn. Other authors use a "find-the-next-comment-and-fix-it" or successive processing strategy immediately.

What strategy an author picks may depend on many factors, such as the stage of the draft and the author's relationship to the reviewer. A system to support annotations, therefore, needs to support both strategies. Authors working with paper annotations can easily scan the margins to get an overall sense of the comments. In many hypertext systems, however, the user must "search and click," that is, the user must search a text node for each annotation and, to display it, must click on a link icon. The annotation is then often displayed out of context, usually in a separate window that may obscure the original text or appear in some arbitrary, distant place on the screen, out of alignment with the original text. A system to support annotations can support the rapid scanning of annotations by providing a constraint-based layout subsystem that displays the annotations on the screen in conjunction with the primary text. Some of the conventions developed by graphic designers for annotating technical drawings might be adapted to solving the problems of how and where to present annotations in node-link graphs so that the annotations are visible "at a glance" and the relationship between the primary node and annotations are easy to see (e.g., by employing "rubber sheet" layout techniques, cf. Kaltenbach, Robillard & Frasson, 23).

In paper documents, annotations from different sources are typically compared by putting two or more pages side-by-side. Unlike paper documents, in electronic documents it is possible to merge annotations from various sources. This possibility raises the requirement that the annotations from various sources remain readily distinguishable even when merged because the source of an annotation can be an important aid in its interpretation. In a linear document, a column for each annotator can be displayed in parallel to the linear text. In a hypertext node-link graph, annotation layers in which color and other typographic features are used to indicate who made the annotation.

Reacting to annotations

With paper, the author must "execute" changes by hand, even when the reviewer has already made the change on the paper copy. With the computer, there is the possibility that the system itself can execute the changes at the author's request (e.g., if the system knows that a particular annotation is, for example a proposed deletion, then at the request of the author, the system can actually delete the text.). InterNote supported this capability by providing the user with a palette of commands ("deletion" "move to" "insert", etc.) so that the user informed the system of the type of each annotation (Catlin, Bush, & Yankelovich, 3). While this represents progress along one dimension over systems that provide no semantics for annotations, it does so at the price of the fluidity with which users can generate annotations. To provide both semantics and fluid generation, a recognition subsystem needs to be able to recognize circling parts of a document (to interpret it as the scope of the annotation), drawing arrows (to interpret it as a "move to" or a link to inserted text) and crossing out text (to interpret it as a text deletion). Without gesture recognition, either (1) the user is forced to inform the system (perhaps via a menu/set of keyboard commands) of the various semantics of the annotations, a situation that lowers the fluidity of generating annotations or (2) the user creates annotations that contain no semantics, in which case the system cannot "execute" the annotation automatically for the author.

With a trusted co-author, an author may be willing to accept all the suggested changes at once. Under normal circumstances, however, an author will want to be able to decide each case. In some cases, authors wish to postpone a decision on a comment. An annotation system can support this activity by providing a function that allows authors to "mark" the annotation as something they want to return to later.

Sometimes a co-author may make a series of interrelated changes. For example, suppose that a co-author moves a paragraph and rewrites the surrounding text to provide transitions to the inserted material and revises the material surrounding the deletion to provide transition to the new context. In such cases, it may be desirable to accept or reject a whole series of changes together rather than piecemeal. To support this, an annotation system needs to provide the concept of a "start-transaction" and "end-transaction" so that a reviewer can indicate that a series of recommendations for change are related. Such a facility also has implications for the display of annotations. The system must highlight the related changes so that an author can see what they are.

Editorial process and annotations

A system to support annotations must support various access rights for reading documents, reading comments (e.g., editors may not want other reviewers to be able to read each others' annotations) and so forth. When providing access rights, it is important to distinguish between rights to write the original document vs. rights to write a version of the document. With documents, it turns out that certain revisions are very difficult to describe and easier to simply fix (e.g., if the tone of a sentence is wrong, it may be easier to fix than to describe). To support a "just do it" strategy observed among expert writers, it is important to allow even reviewers some means of editing the document. Of course, any reviewer must have "read access" to the primary document, so the reviewer could always copy a portion of the document and revise it in the annotation rather than in the text. In a prototype system that implemented this policy, however, reviewers complained about difficulties in getting a sense of the current state of the text when their revisions appeared out of bandwidth from the primary document (Neuwirth, et al., 29). This suggests that a system to support annotations should support versioning and a way to see what changes a reviewer made to the previous version (Haake & Haake, 15; Neuwirth et al., 30).

Editors often want authors to be "accountable" for their recommendations for changes. This is another reason an annotation system needs to support a way of seeing what changed from one version of a document to another. A system to support accountability also needs to support authors being able to annotate annotations, in order to explain, for example, why they did not implement a suggested change or to ask a question about the meaning of an annotation.

Technical documents must often be reviewed by multiple people in an organization: technical, manufacturing, marketing, legal, and so forth. Sometimes people in different organizational roles make conflicting demands for change in a document. A system to support annotations should provide support for negotiating and resolving such conflicts. For example, by allowing the author to flag them as conflicting and sending the relevant sections along with comments from the relevant parties out for re-review. To support conflict negotiation, the system should provide something akin to SEPIA's argumentation space to facilitate the articulation of reasons, counters, and so forth.

SUMMARY AND CONCLUSIONS

In this paper, we have identified issues in computer-supported collaborative writing which have been derived from the task of individual writing, from the process of collaborative writing, and from the need to communicate about writing documents. We then reviewed state-of-the-art research prototypes. We surveyed the SEPIA cooperative authoring environment and showed how it addresses these issues. First, hyperdocuments can be used to create and maintain technical documentation. Second, activity spaces support the cognitive processes of the creation of hyperdocuments. Third, a shared hyperdocument database, versioning, different modes of collaborative work, and group awareness support asynchronous and synchronous distributed collaboration as well as smooth transitions between them. Furthermore, planning structures enables co-authors to coordinate their efforts. Fourth, annotations can be used to comment on drafts and plans.

Although SEPIA in its current form provides strong dedicated support for collaborative writing of hyperdocuments, we identified annotations as one area where further improvement is necessary. Since annotations are one of the major ways authors communicate about documents and current hypermedia systems lack sophisticated support for annotations, we want to continue the line of research that started with the requirements analysis of annotation facilities described in this paper. As a next step, we will design and add features to SEPIA that support generating, receiving and reacting to annotations as well as support annotations in the context of the editorial process.

In order to address the integration of distributed and co-located collaborative authoring, we are about to integrate Xerox PARC's LiveBoard (Elrod et al., 9) with networked workstations for

each participant. In addition to a meeting specific version of SEPIA supporting co-located meetings, we will provide means for integrating it with the distributed SEPIA prototype. Thus, SEPIA can be used to produce documents in pre- and post meeting phases that can be used and changed in the meeting and notes created in a meeting can be used in the post-meeting phase for further document production.

As has been shown in this paper, hypermedia provide benefits for both the product and the writing process. Technical documentation publications may benefit from new media and new organizational structures. Advanced cooperative hypermedia authoring systems can ease the process of collaboratively writing a hyperdocument by supporting coordination, consistent creation and maintenance of large corpora of information in a distributed work setting.

REFERENCES

1. Aksyn, R. M., McCracken, D. L., & Yoder, E. A. (1988). KMS: A distributed hypermedia system for managing knowledge in organizations. *Communications of the ACM*, 31(7), 820-835.
2. Baecker, R. M., Nastos, D., Posner, I. R., & Mawby, K. L. (1993). The user-centered iterative design of collaborative writing software. In *Proceedings of the InterCHI'93* (pp. 399-405), Amsterdam, Netherlands, April 26 - 29, 1993, ACM Press.
3. Catlin, T., Bush, P., & Yankelovich, N. (1989). InterNote: Extending a hypermedia framework to support annotative collaboration (pp. 365-378). In *Proceedings of the 2nd ACM Conference on Hypertext (Hypertext'89)*, Pittsburgh, PA, November 5-8, ACM Press.
4. Chalfonte, B.L., Fish, R.S., & Kraut, R.E. (1990). Expressive richness: A comparison of speech and text as media for revision (pp. 21-26). In *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW '90)*, Los Angeles, California, October 7-10, ACM Press.
5. Crowley, T., Baker, E., Forsdick, H., Milazzo, P., & Tomlinson, R. (1990). MMConf: An infrastructure for building shared applications. In *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW '90)* (pp. 329-342), Los Angeles, California, October 7-10, ACM Press.
6. Dourish, P., & Bellotti, V. (1992). Awareness and coordination in shared workspaces. In *Proceedings of the ACM 1992 Conference on Computer Supported Cooperative Work (CSCW '92)* (pp. 107-114), Toronto, Ontario, November 1 - 4, 1992, ACM Press.
7. Ede, L., & Lunsford, A. (1990). *Single texts-plural authors: Perspectives on collaborative writing*. Carbondale, IL: Southern Illinois University Press.
8. Ellis, C., Gibbs, S.J., & Rein, G. (1991). Groupware: Some issues and experiences. *Communications of the ACM*, 34(1), 39-58.
9. Elrod, S., Bruce, R., Gold, R., Goldberg, D., Halasz, F, Janssen, W., Lee, D., McCall, K., Pedersen, E., Pier, K., Tang, J., & Welche, B. (1992). Liveboard: A large interactive display supporting group meetings, presentations and remote collaboration. In *Proceedings of CHI'92* (pp. 599-607), ACM Press.
10. Fish, R S., Kraut, R. E., Leland, M. D. P., & Cohen, M. (1988). Quilt: A collaborative tool for cooperative writing. *SIGOIS Bulletin*, 9(2&3), April & July.
11. Glushko, R. J. (1989). Design issues for multi-document hypertexts. In *Proceedings of the 2nd ACM Conference on Hypertext (Hypertext'89)* (pp. 51-60), Pittsburgh, PA, November 5-8, 1989, ACM Press.
12. Greenberg, S., & Bohnet, R. (1991). Group Sketch: A multi-user sketchpad for geographically distributed small groups. In *Proceedings of Graphics Interfaces'91* (pp. 207-215), Calgary, Alberta, 5-7 June.

13. Group Technologies, Inc. (1990). Aspects: The first simultaneous conference software for the Macintosh. Arlington, VA: Group Technologies, Inc.
14. Haake, J. M., & Wilson, B. (1992). Supporting collaborative writing of hyperdocuments in SEPIA. In *Proceedings of the ACM 1992 Conference on Computer Supported Cooperative Work (CSCW'92)* (pp. 138-146), Toronto, Ontario, November 1-4, 1992, ACM Press.
15. Haake, A., & Haake, J. M. (1993). Take CoVer: Exploiting version support in cooperative systems. In *Proceedings of the InterCHI'93* (pp. 406-413), Amsterdam, Netherlands. April 26 - 29, 1993, ACM Press.
16. Halasz, F. G. (1988). Reflections on Notecards: Seven issues for the next generation of hypermedia systems. *Communications of the ACM*, 31(7), 836-852, July.
17. Hannemann, J., Thüring, M., Haake, J. M. (1993). Hyperdocument presentation: Facing the interface. *GMD Arbeitsbericht No. ???*, St. Augustin. Also submitted to HCI Journal.
18. Hahn, U., Jarke, M., Kreplin, K., & Farusi, M. (1989). CoAUTHOR: A hypermedia group authoring environment. In *Proceedings of the 1st European Conference on Computer Supported Cooperative Work (ECSCW '89)*, Gatwick, U.K., September 13-15, Slough, UK: Computer Sciences House.
19. Hayes, J.R., & Flower, L.S. (1980). Identifying the organization of writing processes. In L.W. Gregg and E.R. Steinberg (Eds.), *Cognitive processes in writing* (pp. 3-30). Hillsdale, NJ: Lawrence Erlbaum.
20. Irish, P. M., & Trigg, R. H. (1989). Supporting collaboration in hypermedia: issues and experiences. *Journal of the American Society for Information Science*, 40(3), 192-199.
21. Ishii, H., & Arita, K. (1991). ClearFace: Translucent multiuser interface for TeamWorkstation. Research report, NTT Human Interface Laboratories, Kanagawa, Japan, January. Submitted to ECSCW'91.
22. Ishii, H., Kobayashi, M., & Grudin, J. (1992). Integrating inter-personal and shared workspace: ClearBoard design and experiments. In *Proceedings of the ACM 1992 Conference on Computer Supported Cooperative Work (CSCW'92)* (pp.33-42), Toronto, Ontario, November 1 - 4, ACM Press.
23. Kaltenbach, M., Robillard, F, & Frasson, C. (1991). Screen management in hypertext systems with rubber sheet layouts. In *Proceedings of the 3rd ACM Conference on Hypertext (Hypertext '91)* (pp. 91 - 105), San Antonio, TX, December 15 - 18, 1991, ACM Press.
24. Kraut, R.E., Galegher, J., Fish, R., & Chalfonte, B. (1992). Task requirements and media choice in collaborative writing. *Human-Computer Interaction*, 7, 275-407.
25. Kunz, W., & Rittel, H. (1970). Issues as elements of information systems. *Working paper 131*. Berkeley, CA: University of California, Center for Planning and Development Research.
26. Lu, I., & Mantei, M. (1991). Idea management in a shared drawing tool. *Proceedings of the 2nd European Conference on Computer Supported Cooperative Work (ECSCW '91)*, (pp. 97-112), 24-27 September 1991, Amsterdam, Netherlands, Dordrecht: Kluwer Academic Publishers.
27. McGuffin, L., & Olson, G. M. (1992). ShrEdit: A shared electronic workspace. *Technical Report No. 45*, University of Michigan, Cognitive Sciences and Machine Intelligence Laboratory.
28. Nelson, T. H. (1967). Getting it out of our system. In G. Schechter (Ed.), *Informational retrieval: A critical review*. Thompson Books: Washington D.C.
29. Neuwirth, C. M., Kaufer, D. S., Chandhok, R., & Morris, J. (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), Los Angeles, California, October 7-10, ACM Press.

30. Neuwirth, C. M., Chandhok, R., Kaufer, D. S., Erion, P. Morris, J. Miller, D. (1992). Flexible diff-ing in a collaborative writing system. In *Proceedings of the Fourth Conference on Computer-Supported Cooperative Work (CSCW '92)* (pp. 147-154). Toronto, Ontario, November 1-4, 1992, ACM Press.
31. Nunamaker, J. E, Dennis, A. R., Valacich, J. S., Vodel, D. R., & George, J. E (1991). Electronic meeting systems to support group work. *Communications of the ACM*, 34(7). 40-61.
32. Rein, G. L., & Ellis, C. A. (1991). rIBIS: A real-time group hypertext system. *Int J Man Machine Studies*, 34(3), 349-368, March.
33. Schütt, H., & Haake, J. M. (1993). Server support for cooperative hypermedia systems. In H. P. Frei and P. Schauble (Eds.), *Hypermedia* (pp. 45-56). Berlin: Springer-Verlag.
34. Smith Catlin, K., & Garrett, L.N. (1991). Hypermedia templates: An author's tool. In *Proceedings of the 3rd ACM Conference on Hypertext (Hypertext '91)* (pp. 147-160), San Antonio, Texas, December 15-18, 1991, AMC Press.
35. Sproull, L., & Kiesler S. (1991). *Connections*. MIT Press: Cambridge, MA.
36. Stefik, M., Bobrow, D. G., Foster, G., Lanning, S., & Tatar, D. (1987). WYSIWIS revised: Early experiences with multiuser interfaces. *ACM Trans Office Information Systems*, 5(2), 147-167, April. An earlier version appeared in CSCW '86.
37. 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), Pittsburgh, PA, November 5-8, 1989, ACM Press.
38. Streitz, N.A., Haake, J. M., Hannemann, J., Lemke, A., Schuler, W., Schütt, H., & Thüring, M. (1992). SEPIA: A Cooperative Hypermedia Authoring Environment. In *Proceedings of the ACM European Conference on Hypertext (ECHT 92)* (pp. 11-22), Milano, Italy, November 30 - December 4, 1992, ACM Press.
39. Thüring, M., Haake, J. M., & 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), San Antonio, TX, December 15-18, 1991, ACM Press.
40. Todinao, G. (1986). *Using UUCP and USENET: A nutshell handbook*. Newton, MA: O'Reilly and Associates.
41. Toulmin, S. (1958). *The uses of argument*. Cambridge: Cambridge University Press.
42. Walker, J. H. (1989). Authoring tools for complex document sets. In E. Barrett (Ed.) *The society of text* (pp. 132-147). Cambridge, MA: MIT Press.
43. Watanabe, K., Sakata, S., Maeno, K., Fukuoka, H., & Ohmori, T. (1990). Distributed multi-party desktop conferencing system: MERMAID. In *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW '90)* (pp. 27-38), 7-10 October, Los Angeles, CA, ACM Press.
44. Wilson, B. (1992). WSCRAWL 2.0: A shared whiteboard based on X-Windows. *Technical report*, Apple Computer, Mail Stop 35-RJ, Cupertino, CA.
45. Zellweger, P. T. (1989). Scripted documents: A hypermedia path mechanism. In *Proceedings of the 2nd ACM Conference on Hypertext (Hypertext '89)* (pp. 1-14), Pittsburgh, PA, November 5-8, 1989, ACM Press.