

To appear in *Coordination Theory and Collaboration Technology* (Vol. in the Computers, Cognition, and Work Series) by Gary M. Olson, Thomas W. Malone, and John B. Smith (Editors). May, 2000. Lawrence Erlbaum Assoc.

Computer Support for Distributed Collaborative Writing: A Coordination Science Perspective

Christine M. Neuwirth
David S. Kaufer
Ravinder Chandhok
James H. Morris

Carnegie Mellon University
Pittsburgh, PA 15213

GOALS OF THE RESEARCH

The goal of our research is to provide computer support for distributed collaborative writing. Writers can be said to be distributed when they have distributed knowledge and skill, and they share that knowledge and skill in order to develop a draft; or, even when they have significant overlap in knowledge and skill, they distribute the work of producing the draft itself among them. But in this sense, all collaborative writing is distributed. In the sense we will use the term here, distributed collaborative writing refers to, additionally, situations in which the writers are distributed in time (i.e., they do not work on the artifact at the same time) or place (i.e., they do not meet face-to-face). The central research questions in distributed collaborative writing are "What does the process of producing a written product look like when it is divided among writers who coordinate to produce it over time and space?" and "What is the relationship of these processes to success?" When the process includes "active agents," the scope of the first question shifts slightly to include not only people, but computers as well. This question is, of course, the central question of "distributed cognition" or "coordination science," applied to collaborative writing. In analogy with the way cognitive scientists (psychologists, AI researchers, etc.) are interested in identifying strategies and representations involved in individual cognition, coordination scientists are interested in identifying the strategies and representations that groups of "agents"--people and computers--use to coordinate their activities (Malone, 1988).

The central question of computer support for distributed collaborative writing is "What are the requirements for supporting distributed collaborative writing processes?" This question is related to the previous questions: In order to produce knowledge that is useful in designing computer support, the description of strategies and representations needs to be sufficiently detailed that it yields answers to a set of related questions. These questions include (a) What problems do such writers have and are there ways computers can mitigate them? (b) Are there ways that computers can augment the processes? (cf. Olson & Olson, 1991).

There are properties of the process of writing that makes it an interesting, though challenging, application domain for coordination science. Writing is an open-ended design process. A *design* process is one that involves the creation of an artifact. An *open-ended* design process is one in which any existing specifications for the artifact leave many design decisions open, design decisions that nevertheless must be made in order to create it. Moreover, any specifications that might exist are often open to interpretation, and the more heterogeneous the background knowledge and skills of members of the collaborative writing group, the more likely that differences in interpretation will arise (Gabarro, 1987). As a result of these properties, there may be situations in which members of the collaborative writing group do not have shared knowledge, shared goals or criteria, or even a shared representation of how best to proceed with the task (cf. Hewitt, 1986).

OUR RESEARCH STRATEGY

Our research strategy can be outlined by the following steps (Neuwirth and Kaufer, 1992):

- Identifying writers (e.g., novices, experts) and a writing task (e.g., co-authoring). In our research, we have not focused on collaborations in which co-authors or commenters interact face-to-face, though systems that support research into the issues such collaborations raise are clearly valuable (McLaughlin Hymes & Olson, 1992; Olson et al., 1992).
- Building a theory- and research-based model of the task, with a focus on user-centered design (Gould, 1988). Understanding how writers function and hypothesizing the sources of their successes and failures is vital to building tools to support writers. The model draws upon techniques, both cognitive and social, for building models of composing processes, but focuses on *problems* that writers--even experienced ones--have with the task. This model informs the design of technology.
- Designing technology to alleviate these problems. This step involves building a theory of the *prima facie* ways computers can augment writers' performance of the task by drawing upon a theory of the role of external representations and a theory of task activity. The technology represents a hypothesis about a solution, perhaps a partial solution, to some needs or problems identified by the theory- and research-based model. We have embodied our theories of distributed collaborative writing into a "work in preparation" (PREP) editor, a multi-user environment to support a variety of collaborative and, in particular, co-authoring and commenting relationships for scholarly communication.
- Studying the technology in use, with the aim of building knowledge that will help to refine the model of the task and the design of the technology.

These steps are interconnected and often recursive. For example, studying one of our software tools, the Comments program, in actual use led us to refine our model of the task and to design a new software tool, the PREP Editor (Neuwirth, et al., 1988; 1990). Indeed, given that all writing involves technology (e.g., pen and paper), the last step can be thought of as the second step repeated. It is not necessary for the steps to be carried out by the same group of researchers. A study by a group of empirical researchers observing a software tool in use may be relevant to researchers working at other steps, perhaps on the theory of composing or on the design of software. For example, the work of Haas and Hayes (1986; 1989a,b), which identified the problems writers have of getting a "sense" of their texts when using word processors, added to our theoretical understanding of the process of composing by identifying an additional subprocess, the subprocess of reading one's own writing, and highlighting its importance. This result stimulated further research into the role of reading during writing, both in print and hypertext environments, and has been used by our research group to inform software design.

While it is not necessary, then, for the steps to be carried out by the same group of researchers, it is necessary, or at least desirable, that researchers understand the interconnectedness of the steps in order to increase the likelihood that results they produce at one step will be relevant to other steps.

THE PROCESS OF COLLABORATIVE WRITING

In the following, we outline our model of collaborative writing. In its major outlines, we have drawn heavily upon the process model of writing developed by Flower & Hayes (1981a; Hayes & Flower, 1980). Although developed for single authors, the empirically-based model is also a useful starting point for characterizing the cognitive processes involved in collaborative writing and, supplemented by observations about actual collaborative writing groups, in deriving design

requirements for computer support for those processes. In the following, we introduce the major components--planning, drafting, and reviewing--of the model, discuss some implications of the model for design requirements, and provide examples of how our prototype attempts to meet those goals.

In order to make general observations about collaborative writing concrete and to gain further insight into how to support groups, we observed groups of writers working under the following conditions: (1) able to meet face-to-face, (2) able to work at the same time but not face-to-face, and (3) neither able to meet at the same time or place. The groups, consisting of three writers each, were asked to write a press release, respond to two letters, and write a brief report on their activities.¹ In the following, we describe PREP Editor's support for collaborative writing by drawing upon observational data from one of these groups, the group that was neither able to meet at the same time nor place.

Planning

Planning refers to processes of generating (1) criteria for the text (e.g., the purpose of the text, features the text needs in order to meet the needs of its audience, etc.), (2) ideas for the content of the text, (3) plans for how to organize that content, and (4) plans for how to proceed with the process of writing itself (e.g., deciding that particular people will write particular parts of the document or do particular tasks such as review for technical accuracy or style).

When writers work alone, they may not need to articulate the constraints that they have imposed and the goals they have set (though studies of experienced writers working alone indicate that they do, indeed, record *some* of their plans). Not surprisingly, co-authors often need to communicate about plans in order to refine their views of the goals that co-authors have generated and increase the likelihood that they will generate compatible products.

From a coordination science perspective, co-authors (or co-authors and reviewers) must manage a producer/consumer relationship--that is, whatever is produced should be usable by the activity that receives it (Malone & Crowston, 1994). This producer/consumer relationship applies not only to the usability of the final draft for readers, but also to the usability of intermediate drafts that co-authors/reviewers exchange among themselves. Communication about plans, goals and constraints may improve usability by saving co-authors and commenters from having to infer the other's plans. If other co-authors understand the goal, they may be more likely to be able to produce revisions to the draft that are compatible with the first author's goals, and the draft they produce is more likely to be seen as useful by the other author. Or if another author has a different point of view, it may be more likely to surface and be resolved. Of course, unnecessary communication can also be distracting, leading to degradation in performance. Research we are currently conducting attempts to relate such patterns of communication to measures such as the time to complete a project and the quality of the product.

When collaborative writers are able to meet face-to-face, they can communicate about plans relatively easily: Face-to-face communication is both highly interactive (e.g., requests for clarification can be answered immediately) and expressive (e.g., facial cues and gestures can also be used to communicate). Even face-to-face communication, however, is not without its problems and, interestingly, may be augmented by computer support (Olson et al., 1992). But writers seem to experience more difficulties when working over distances. In a study of groups of writers

¹We studied the group's activities and interaction behavior by observing and videotaping their meetings asking subjects working alone to think-aloud. The group members were PhD students in English.

working face-to-face vs. at a distance, for example, Galegher & Kraut (1994) report that writers working at a distance with traditional computer-mediated communication tools (e.g., email and conferencing tools) and phone needed to spend more time to achieve the same quality of result and reported less satisfaction with their work and with other members of the group than those working face-to-face. Two recent lines of research represent attempts to mitigate these problems. The first, shared editors supplemented by audio (e.g., phone) and video links, allow writers to communicate at the same time over distances, supported by the ability to see an evolving draft. The PREP Editor prototype can approximate synchronous communication through a set of parameters that allow writers to control how quickly replicated copies are transmitted to others (Neuwirth, et al., 1994). The second line of research focuses on writers working over distances asynchronously. We have extensive experience with our prototype for supporting the latter, and will focus on ways we have seen it used here.

A collaborative writing group can use PREP Editor's column interface to discuss initial design decisions. Figure 1 shows a display in which one writer has formulated a set of questions about the writing project as a whole in the leftmost column; the second and third columns, which are linked to the first, depict responses from the other writers. While collaborative writers can use the PREP Editor interface to discuss such overall design decisions and we have observed them doing so spontaneously, an important outstanding research question remains whether this interface is effective in facilitating such discussions. Some groups we have observed continue to use email to conduct such discussions. Whether this is a matter of adaptation remains to be seen. The groups we observed were not shown this possible use of the tool, though we are conducting an observational study in which this use will be demonstrated.

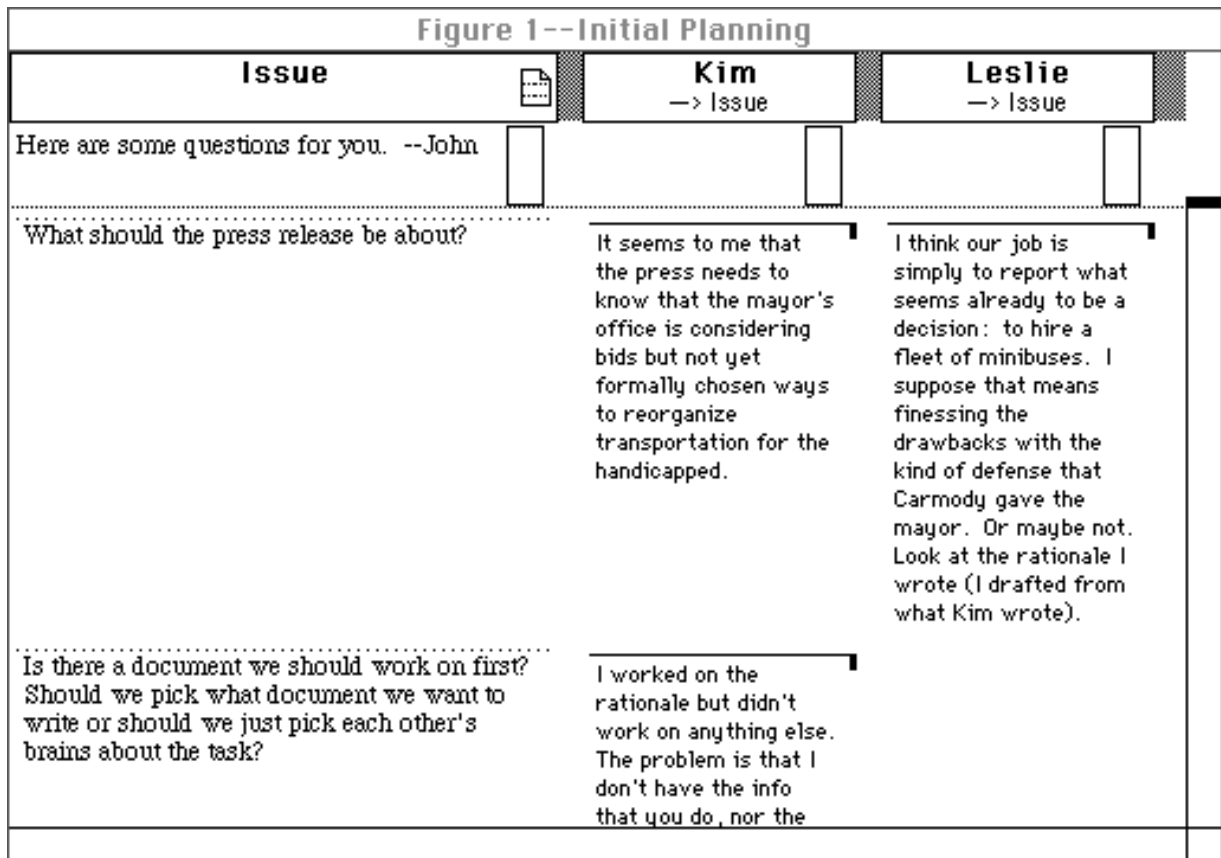


Figure 1: Communication about initial design decisions in the PREP Editor.

Groups of collaborative writers using the PREP Editor over distances use the tool frequently and spontaneously to communicate about plans, goals and constraints in a way that is grounded in an evolving draft. As an example of such communication, Figure 2 depicts the draft of a letter, together with the author of the draft's comments on the draft, intended for the other writers and explaining the goal of the paragraph. (Note also, the author indicating a judgement about the completeness of, and confidence in, parts of the document).

Figure 2--Goal Communication	
Letter	Leslie's comments -> Letter
<p>option to adopt a volunteer transport service.</p> <p>.....</p> <p>Nexus decided to support the minibus service because that option is the only customized transportation service that meets the special needs of the differentially abled. Although problems, such as those you outlined in your letter, will have to be addressed as this new system is put into place, we believe that many of these problems can be anticipated and resolved.</p> <p>.....</p> <p>I am forwarding a copy of your letter to the city planning office along with our proposed plan. Whether the final decision of the</p>	<p>I'm not sure this paragraph is even worth keeping, but I thought it might just give a little more in-depth rationale for proposing minibuses. But I'm not adverse to zapping this if it just sounds like we're rehashing the same info.</p>

Figure 2: Communication about plans, goals and constraints in the PREP Editor.

Although there is a tendency to equate the act of writing with producing the content of the written draft, studies show that experienced writers typically engage in many acts of writing (e.g., jotting down ideas, drawing) that bear no direct relation to the text product, but serve as inexpensive, intermediate external representations to remind writers of their plans for audience, purpose, and procedure, as well as content (Flower & Hayes, 1989; Haas, 1990). When working with computer environments that do not support the creation of arrows, boxes, or other diagrams for displaying conceptual relationships among ideas and the suppression of detail, writers report frustration (Bridwell-Bowles, et al., 1987) and important planning activity is curtailed (Haas, 1989b). Thus, observations of expert writers at work suggest that supporting the jotting, drawing and note-taking that writers engage in as they write are especially important in writing and that cognitive aspects must be taken into account when designing computer support for co-authoring and commenting tools. There have been some attempts to understand the task-specific activities (e.g., jotting, drawing, writing, gesturing) that occur in collaborative tasks in order to inform the design of specialized tools to support those tasks (Stefik, et al., 1987; Tang & Leifer, 1988). But because there is a tendency to equate the substantive work of writing with a written draft, most text annotators support only communication about the working draft or outlines of a draft. The PREP Editor contains a drawing tool and the objects produced in the drawing can be annotated, but it has not been used extensively to support planning by groups we have observed. An outstanding research question is to what extent this may be due to the fact that the very rough sketches and private jottings that writers working alone produce are less "sharable" artifacts and to what extent it is a deficiency in the tools to facilitate such informal sketching. An interesting research approach might be to provide tools for graphically-based idea generation, roughly corresponding to the "network mode" in Smith *et al.*'s Writing Environment (1987), along with pen-based input.

Drafting

Drafting is the process of producing text. Studies of experienced writers indicate that they often set new goals for themselves as they draft, that is, they discover what it is they want to say in the process of saying it (Hayes & Flower, 1980). As a result of this property of writing, a collaborative writer's knowledge of other participants and their actions may be uncertain or changing. This has been confirmed by case studies of collaborative writers at work. For example, Kaye (1993) observes:

Regardless of the level of detail in a specification, more often than not, the author's ideas only become clear when the extended draft is being written. As a result, colleagues' perceptions of what is being written (based on an earlier draft) may not be an accurate reflection of what any individual team member is in fact writing. Riley (1984) refers to this as the 'out-of-step' phenomenon. Various informal tactics are used to minimize the likelihood of [integration] problems arising, and this is obviously fairly straightforward when colleagues work in adjacent offices and see each other regularly, both formally and informally, in between the meetings.

Any system to support collaborative writing needs to accept that any plans made in advance of drafting do not control completely drafting, indeed, that plans will not be made completely in advance of writing and must support communication about changes in plans. Groups using PREP Editor use the interface to discuss evolving plans frequently.

A second property of drafting is that the partially completed product plays an important role in open-ended design processes: The partially completed product becomes part of the task environment and constrains the subsequent course of the design. Writers frequently re-read portions of the text they have produced so far to provide constraints for another segment of text that they want to produce (Flower & Hayes, 1981; Kaufer, Hayes, & Flower, 1986). From a coordination science perspective, the draft itself is a "shared resource" that all writers may benefit from accessing, even if they have agreed to work on a particular part.

These two properties of drafting suggest that there will be situations in which collaborative writers could benefit from having parts of the document that others are working on available. This observation must be tempered, however, by noting the need to take the wide variability of writing groups into account. Some writing groups want updates to drafts to be available immediately to all members of the group. Other groups want information about changes delayed until the source has been able to check them for correctness and "commit" to them. For example, Newman and Newman (1993) describe a case study of a large group of writers working on a document to support budget allocation decisions within their organization. Different departments had responsibility for different parts of the document, and the outcome of the budget allocation would affect their respective budgets. In this case, departments concealed early drafts from members of other departments, so that the text would not be available to writers outside the department until the political issues had been thought through by writers inside the department. Likewise, individual authors vary considerably in their styles of working (Posner & Baecker, 1992), with some authors anxious to receive immediate feedback on even "half-baked" ideas, whereas other authors, as Kaye (1993) observes, do "their writing in concentrated bursts of activity prior to previously agreed deadlines, and, in any case, may not wish to make their developing drafts public."

These observations suggest that a system to support collaborative writers should provide authors with the ability to be flexible in making parts of documents visible. We have defined a set of parameters of interaction for the networked version of our prototype, parameters that allow writers flexibility in sharing partial results along the dimensions of who to share with, and what, when and how quickly to share (Neuwirth, et al., 1994). Each user can set these parameters to define his or her own pattern of data exchange. For example, setting parameters for data to flow automatically at a grain size of a keystroke with fast transmission speed approximates synchronous

communication. On the other end, setting parameters for data to flow only upon explicit request at a grain size of the column models the situation in which a co-author requests to see the latest version of another co-author. By setting these parameters, users can adjust the characteristics of the information flow.

A good deal of the work has been directed at mapping flexible social protocols onto practical communication protocols. Clearly, if social protocols are to be flexible, data exchange protocols (within a system) and network protocols (across systems) must be as well. As far as the flexibility of data exchange within a system is concerned, Dewan and Choudhary (1991) have argued that systems must be flexible in their assumptions about data interaction. They have proposed a set of system primitives that would allow users to calibrate their assumptions about the exchange of information (not simply data, but views, formats, and windows as well) to other users in a flexible fashion. Our work builds on some of their primitives by allowing for the incremental vs. complete exchange of data, but extends it for collaborative writing applications. Future work needs to extend this to sharing views and windows as well.

The fact that different writing situations require integration of asynchronous and synchronous styles of work has also been noted (Dourish & Belotti, 1992; Posner & Baecker, 1992). Minör & Magnusson (1993) present a system to support an integration of writers' asynchronous and synchronous work strategies. Their model is similar to the one described here, in that it is based on working with copies of a document rather than actually sharing a document. It differs in that, when a user opens a version of a document that is currently being edited by another user, the system attempts to make users aware of each other's activities by showing the differences between two versions of the document. Other parameters of interaction (grain size, flow, etc.) are not defined. The underlying model relies on system knowledge of the hierarchical nature of documents (sections, subsections, etc). In contrast, the model described here requires minimal interpretation of the document structure per se (import/export algorithms that support reading documents produced in other word processors interpret paragraph breaks as chunks).

Up to this point, we have only discussed collaborators *viewing* sections of documents that are being worked on by others. There is reason to suppose, however, that it can also be useful for collaborators to be able to change sections of documents being worked on by others. Because texts have "texture," that is, coherence relations throughout, a change in one section may require changes in another. That is, although writers may decompose writing the document into subtasks, the decomposition almost always entails interdependencies among subtasks. While it is possible for an author to suggest a change to an author of another section, it is sometimes more efficient to simply "do it." In our model, co-authors can choose to accept a non-conflicting change automatically or to receive notification of the change with final approval residing with the co-author responsible for the section.

Reviewing

The process of reviewing consists of two subprocesses: evaluating text and revising text. Often this evaluation and review process takes the form of comments on the text. The problems with comments, that is, critical notes on texts, are well-known and legion: writers don't understand comments, they think the comments reflect confused readings rather than problems in their texts, they are frustrated by perceived lack of consistency in comments and contradictory comments (Neuwirth, et al., 1988). The problems in author/commenter relationships become even more pressing if authors solicit comments from multiple readers. From a coordination science perspective, there is also a consumer/producer relationship that must be managed in the review process. When collaborative writers can meet face-to-face, this relationship can be managed by communication about the comments. For example, in an observational study of a group of physicists working to produce an article over a period of months, Blakeslee (1992) observed face-to-face meetings in which members of the group discussed and clarified comments that they had

made on drafts. This suggests that computer support for distributed collaborative writing should support discussion, not only about plans and drafts, but also about the comments themselves.

A second form that evaluation and review takes, at least among co-authors, is actually making changes to parts of a document that someone else has written. A principal difficulty co-authors face is in coping with those changes, especially understanding why the other person made them. For example, in a study of eight writers' production of an insurance company's two-page annual report, Cross (1990) observed that each writer "omitted, added, highlight or modified" the text to agree with his or her preconception, with unexplained changes causing "considerable frustration" for other writers and an undetected change causing a major problem (p. 193).

This suggests that a system to support collaborative writing should support the detection of changes from one version to another, along with supporting communication about those changes (e.g., annotating the changes with questions about the decision and explanations of changes).

With paper documents, even reviewers often make "changes" in content by marking up the draft. This phenomenon may be due to the fact that many significant problems in texts (e.g., voice, persuasiveness, and organization), though easy for an experienced writer to detect, cannot be easily described. For such problems, rewriting is often a more efficient strategy than trying to describe the problem, and writers often choose this strategy when revising others' texts (Hayes, et al., 1987). Some early systems to support collaborative writing (Comments, Quilt) restricted reviewers to the role of attaching comments to the base document. While this increases the usability of the commenters' activities from the point of view of the author, it seems to increase the difficulty of the task for reviewers. In our observations of reviewers working with the Comments prototype, writers in the role of commenters often copied a region of the base document into a commenting box and proceeded to rewrite the copy. Writers who worked in this fashion, however, reported difficulties in revising because their revisions were physically separated from the larger body of text. More specifically, they reported needing a "sense of the whole text" even when commenting on a part. One exasperated commenter went so far as to copy an *entire* document into a comment box and to revise it from there. Whether a commenter is able to modify the base document or not should certainly depend on his or her rightful relationship (co-author, commenter) to the text. Despite potential problems, role specification is likely to be a useful strategy for managing some coordination problems; our design, however, allows new ways of dealing with this interdependency by giving commenters the ability to rewrite his or her *view* of the text and supporting ways for authors to see the changes as proposed changes to the original base document.

Figure 3 depicts an interface for detecting changes from one version of a draft to another in the PREP Editor. The comparison interface produces its report in a new column, with the differences linked to the original column for easy, side-by-side evaluation. To illustrate, Figure 3 depicts four columns: an original draft, its revision, the comparison report and an evaluation column. An author can "push" particular revisions across the link, as in InterNote (Catlin, Bush & Yankelovich, 1989). The evaluation column in Figure 4 consists of annotations to the comparison report that a co-author produced in order to explain some of the changes or to solicit advice about them. An important feature of the interface is the ability of users to annotate changes with explanations of the change or questions to co-authors. Our group's experience with this feature suggests that reviewers will annotate changes selectively in order to draw their co-authors' attention to changes they want to discuss or explain. Likewise, a co-author can ask a reviewer to explain a change. Our hypothesis is that the ability to annotate changes will greatly alleviate writers' frustrations with undetected and unexplained changes that Cross (1990) observed.


Figure 3--Changes to Drafts			
Press Release 	Press Release, -> Press Release	Comparison -> Press Release	John's -> Press
proposing that the city modify and reinstate the minibus transportation system that was discontinued two years ago. According to one member of this team, "We decided to go with the minibus service because that option is the only one customized to the special needs of these physically disabled citizens. We know about the problems that existed in the past, but we feel many of these problems can be anticipated and resolved."	proposing that the city modify and reinstate the minibus transportation system that was discontinued two years ago. According to one member of this team, "We decided to go with the minibus service because that option is the only one customized to the special needs of these physically disabled citizens. Any problems experienced in the past with minibuses have been anticipated and will be resolved."	<p><u>REPLACED "We know about the problems that existed in the past, but we feel many of these problems can be anticipated and resolved."</u></p> <p><i>WITH "Any problems experienced in the past with minibuses have been anticipated and will be resolved."</i></p>	I like this as it is. I made some changes in wording in the last sentence.

Figure 3. Communicating about Changes in Drafts

We have experimented with heuristics for automatically generating comparison reports, depending on role relationships among writers. For example, if the annotated draft is from a co-author, then display changes upon request; if from a reviewer, then display all changes automatically. Apart from generating the comparison before returning the revision (which we, as co-authors, currently sometimes do), the revision's author has little control over how the comparison might be done. As this information might lead to a more productive exchange, we plan to experiment with adding "comparison settings" information to revisions that would serve as hints from the co-author to anyone who would generate a difference report.

THE PREP EDITOR PROTOTYPE

The PREP editor² prototype, then, embodies part of our theory of collaborative writing. It approaches requirements for by supporting communication about plans, constraints, drafts, comments, etc. and by providing a flexible set of parameters for interaction. Central to the PREP editor is a focus on providing a usable, visual representation of the information that will allow new

²The prototype runs on Macs and is available by anonymous ftp at [english.hss.cmu.edu/english.server/mac/prep-editor](ftp://english.hss.cmu.edu/english.server/mac/prep-editor) or <ftp://english.hss.cmu.edu/english.server/mac/prep-editor/>.

kinds of ways of managing interdependencies in open-ended design tasks, in addition to supporting existing patterns.

The Interface

Much of our work has focused on the interface, specifically on the visual representation of the draft and an optimized action grammar. For the visual representation, we have pursued a path that could be called "dynamic glossing," since we support annotation in a style similar to old, glossed scholarly texts. While in some sense this means that we are mimicking the static annotation process, we are also taking advantage of the dynamic nature of the computer to use visual cues such as shading and spatial relationship to show the interconnections among chunks in the system. To create a visual system that will lend itself to providing and accessing comments easily, the visual grammar must be capable of supporting writers' needs. We have found, for example, that visual alignment of comments is a useful feature for allowing collaborators to see comments "at a glance" (see Figure 1), but in a flexible system, the general case requires a constraint-based layout algorithm that can handle arbitrary shapes and complex interconnections among dynamically selected items (Smolensky, et al., 1987). We have also worked on the action grammar, optimizing actions that are used frequently. For example, to create a comment, a writer need only click and drag the mouse.

The cognitive needs of collaborative writers are too numerous to detail here. We focus, therefore, on one: accessing comments. Most text annotation systems are based on a hypermedia model and the primary method for accessing information in hypermedia systems is following link icons from node to node. Typically the user brings a node (e.g., a text node) onto the screen, reads its contents and notes any links, then chooses to traverse some of the links. Such localized link following is adequate for browsing tasks but has been problematic for others (Halasz, 1987). For example, we have found that co-authors and commenters want to visually scan a set of comments quickly and resent the time required by the "search and click" interface to call up each comment, inspect it and put it away. Some researchers have worked to tailor the navigational linking system of hypermedia systems to meet user's writing needs (Catlin, Bush & Yankelovich, 1989; Edwards, Levine & Kurland, 1986; Fish, et al., 1988; Neuwirth, et al, 1987), but the access problem remains to be addressed. Our approach calls for a tailoring the program to match user's cognitive activities (Norman, 1986).

We have previously analyzed the design features of twelfth century glossed bibles in Cavalier, et al. (1991) and the match to the cognitive needs of commenters. The analysis suggests the following requirements (see Figure 1):

- *The primary text is easily distinguishable from the annotation text.* This requirement allows readers, who may not have seen either the original text or the annotations, to orient themselves to the texts quickly. In glossed bibles, this distinction was usually made by varying type size: the primary text is several points larger than text in the annotations. Of course, other typographic signals such as color could also be used.
- *The annotations are visible "at a glance" while reading the primary text.* This requirement minimizes the problems readers have in accessing annotations. Glossed bibles were usually the result of calligraphic as well as scholarly effort; the annotations were packed in an aesthetically pleasing fashion onto a page, so that all annotations, no matter how dense, are visible. As the corpus of annotations increased over time, the books were recopied with more space for annotations, preserving the easy access by expanding the leading between the lines in the primary text as needed to insure the visual alignment of all annotations.

- *The relationship between the primary text and the annotations is easy to see.* This requirement insures that readers will be able to see which annotations refer to particular portions of text. In glossed bibles, the annotations were typically aligned horizontally to the primary text, so it is possible to scan from the primary text across to the annotation rapidly. Moreover, the scope of the annotation was usually indicated by graphic symbols in the primary text.
- *Different contributors are readily distinguishable.* This requirement aids readers in interpreting annotations by different commentators. The different annotators of glossed bibles are easily distinguishable because each has his or her own column.

The result of all these features of glossed bibles was that *access* to the annotations was superior to most electronic annotation systems. The reader could quickly skim the set of annotations "at a glance." For the scholar, the assignment of marginal "real estate" allowed for quick and easy annotation. A comment could be made as quickly as moving the pen to the adjacent margin. In addition, several scholars often annotated a document side by side, leading to an easy to follow parallel discussion that was the synthesis of both sets of comments. We have embodied these features in the PREP Editor prototype.

Implementation

The PREP Editor utilizes an underlying node-link architecture. This section describes those mechanisms and discusses the features that underlying hypertext engines need to provide to support annotations as they are implemented in PREP. While the PREP Editor prototype restricts itself to linear texts, the model employed can be applied in those portions of a hypertext applications that provide design objects to support a linear layout of nodes and links, for example, GUIDE , the Rhetorical Space in SEPIA (Haake & Wilson, 1992).

To describe the PREP architecture and interface, we will use terms drawn from the Dexter hypertext reference model (Halasz & Schwartz, 1990). The Dexter model divides hypertext functionality into three layers: *the storage layer*--the node/link network structure; *the runtime layer*--the mechanisms supporting the user's interaction with the hypertext (including presentation); and *the within-component layer*--the content and structure *within* nodes and links. The fundamental entity in the Dexter storage layer model is a component, either an atomic component (or node), a link component (or link), or a composite component (or composite node) composed of other components.

The Column: A Composite Component/Linear Presentation Pair

The PREP Editor defines a *column* to be a composite node consisting of atomic or composite nodes with "path" links between them, forming the nodes into a connected graph. The nodes of a column are further constrained in that the "path" links, together with a traversal mechanism defined for them, must allow the hypertext runtime layer to construct a linear ordering for the nodes (i.e., to display the nodes linearly). The structure of links among the nodes does not *have* to be restricted to a directed-acyclic graph (DAG), but the traversal mechanism must include a decision rule for finitely terminating any cycles.

The Annotation Link: A Binary, Directional, Typed Link

"Annotation links" are binary, directional, typed links from a source node to an annotation node. The PREP Editor allows users to create annotation links between columns (i.e., composite

nodes).³ For example, in Figure 1, the two columns on the right are linked to the leftmost column. Such links define a tree of linked columns (although a single PREP document can hold a forest of these trees). Links between columns allow users to create annotations rapidly: a user only has to locate *where* in the primary text to make an annotation and click next to that location in the linked column to make both a link and new node.⁴ This user interface allows us to approximate the ease of paper-based annotating by proximal writing in a margin. However, PREP goes one step further than a paper-based metaphor in that PREP annotations remain aligned to their source even as the source text is edited.

The Layout Algorithm

In most hypertext systems, a link, regardless of its type, is represented in the runtime layer layout as a line connecting two squares (in a graph view), or is represented by an icon that is given a "follow it" interpretation when the user clicks on it. The links in the PREP application, however, carry different implications for the runtime/presentation layer. Path links between nodes in a column result in a linear, scrollable display of the nodes that look like an ordinary document in a word processor. Annotation links between columns result in the linked column being allocated (by default) a narrower display and smaller font; unlinked columns, which usually contain a primary text, are allocated (by default) a wider display and larger fonts. Annotation links between nodes result in the nodes being displayed in a side-by-side, horizontal alignment.

An object-oriented constraint-based layout algorithm is at the heart of determining and maintaining side-by-side layout of annotations. Whenever a user changes the screen through some operation on a column or node--creating, moving, deleting, linking, adding to and so on--constraints from the local objects on the screen whose display is affected by the change are placed in a queue. A constraint solver satisfies the constraints; their satisfaction often leads to the propagation and satisfaction of more constraints. The process of constraint maintenance and propagation continues cyclically until the display is no longer affected. The PREP editor requires dynamic communication between the within-component layer (the content and structure within nodes and links) and the runtime-layer. In particular, the PREP editor requires *component location and size information*, and *size and location change events*. For example, a constraint such as *AlignToptoTop* requires the constraint solver to be able to query an object about its position on the screen. This, in effect, results in querying a link about the location of the component at the other end.

The Storage Model

The storage model presented to users is a database. Whereas a file system presents data in a file as an uninterpreted byte sequence, a database encapsulates substantial information about the types and logical relationships of data items stored within it. A database model allows users to exploit

³When a user creates a new column, if a column is selected, then the new column will be linked to the selected column. If no column is selected, the new column will be unlinked. To link two existing columns, a user selects the "from" column, chooses "Link" from a "Column" menu, and choose the "to" column.

⁴If the user actually makes a selection in the primary text and clicks in the linked column, then the system puts a link anchor around the selection; otherwise, the system puts a zero-length link anchor in the same line as the primary text.

knowledge of the data stored in it (e.g., chunks, linked columns, etc.) to achieve better performance.

Information Transport Architecture

The abstract system architecture consists of three logical units: a source, a filter, and the receiver(s). In this system, a source agent consults the filter to determine what information to send to which receivers. The work has significant similarities with work in information filtering (cf. ACM Special Issue, 1992). The differences reside in the information source having human agency and thus being able to offer information based on non-formalized models of receivers' interests and states.

Information transport can be done in a variety of ways. We have chosen to emphasize scalability and independence from details of an indigenous network in our implementation. We want the system to support small user groups of two to four individuals, as it has in the past. But we also want it to scale upward gracefully to much larger groups across wide-area networks. A goal, for example, is to have PREP support users of major national research and information networks. As a result, we have chosen stochastic (rather than deterministic) algorithms, of the kind represented in epidemic algorithms (Demers et. al., 1987), as having desirable properties of scalability and independence from many of the details of an indigenous network.

The highly replicated model we are employing has significant implications for consistency. Issues of concurrency control for advanced applications such as cooperation and coordination are just beginning to be studied (Barghouti & Kaiser, 1991). Most work has been done in the context of software development environments and some in CAD systems. It is important, however, to pursue development of protocols for collaborative writing as well, since the consequences of inconsistency are quite different for different domains (e.g., the object code produced by compilation may be invalid, whereas a writer may be able to regain consistency by reversing the effects of some operations explicitly). Thus, we believe that writers may be willing to trade "high availability" for "accuracy," an issue we intend to explore through providing users with a set of parameters that they control.

This strategy turns "consistency" from an accuracy issue into a performance issue. An example of a system that implements a similar policy is the Coda file system (Kistler & Satyanarayanan, 1991). Except for a small number of files that must remain consistent, Coda's strategy is to provide the highest availability at the best performance: The most recent copy that is physically accessible is always used to satisfy a file request. Coda's view is that inconsistency is tolerable if it is rare, occurs only under conditions of failure, is always detected, and is allowed to propagate as little as possible. Kistler and Satya note that it is the relative infrequency of simultaneous write-sharing of files by multiple users in most file system environments that makes this a viable policy. Unfortunately, the assumption of infrequent simultaneous write-sharing of files sometimes fails to hold for collaborative writing groups. Several users may want to share parts of a document in order to be able to work. For example, in a tight deadline, one person may be working 5 paragraphs behind another, the first person drafting technical details, the second, adding support from empirical research results. While this situation could be accommodated by implementing the artifact as a large number elements implemented as small files, it is easy to imagine situations in which this solution breaks down: for example, a person may, while drafting, discover that he or she needs to make a change to a part of the document currently being worked on by someone else. On the other hand, collaborative writing groups spend large parts of their time engaged in less synchronous activities. A fixed policy that penalizes such groups to accommodate the simultaneous write-sharing case seems inappropriate.

EMPIRICAL STUDIES

To examine the PREP Editor prototype's support for voice modality, we undertook a study with two goals: to compare the nature and quantity of voice and written comments, and to evaluate how writers responded to comments produced in each mode (Neuwirth, et al., 1994). Writers were paired with reviewers who made either keyboarded or spoken annotations from which the writers revised. The study provides direct evidence that the greater expressivity of the voice modality, which previous research suggested benefits reviewers, produces annotations that writers also find usable. Interactions of modality with the type of annotation suggest specific advantages of each mode for enhancing the processes of review and revision. This study adds to the previous picture of the utility of the voice modality for supporting collaborative writing activities. The results can be summarized as follows:

1. The mode of production (keyboarded vs. spoken) affected the type of problem that reviewers communicated: While all the reviewers in the study produced more comments on problems of substance than any other type of problem, reviewers in voice mode were likely to produce more comments about purpose and audience than reviewers in keyboard mode, while reviewers in keyboard mode were likely to produce more comments about substance.

It may be that the written text, which more readily permits review of what has been written, reflection upon it, and revision, may facilitate comments that involve complex substantive issues. If production modality does influence the types of problems communicated, then writing tools offering both modes may need to provide guidelines for choosing the most appropriate mode to work in for encouraging evaluation at the appropriate level.

2. The mode of production affected how reviewers characterized problems. While reviewers in both modalities produced about the same number of annotations overall, the number of words per annotation was far greater in speech. This difference can be accounted for, in part, by the greater frequency of reasons and by the greater number of words used to produce mitigated statements. A higher proportion of the annotations produced in voice contained reasons why the reviewers thought something was a problem and polite language that mitigated the problem.

3. The mode of production affected how writers perceived their reviewers. Writers' evaluations of their reviewers were likely to be less positive when reviewers produced written annotations than when they produced spoken.

4. The study failed to find an overall difference in reviewers' assessments of how responsive writers were to annotations produced or received in the two modalities. Future analyses are planned to examine whether the nature of the annotations and writers' perceptions of reviewers interacted with responsiveness.

5. Despite the previous research findings that spoken annotations would likely be tedious to listen to and more difficult to process, writers using the PREP Editor interface for voice annotations were generally favorably disposed or neutral to voice annotations for most types of comments, except low-level mechanical ones.

In this study, authors chose their reviewers and reviewers were constrained to produce comments in only one modality. More research is needed that varies both conditions of producing annotations and the social relations between the writer and reviewer and looks at annotation interfaces for other sorts of documents (e.g., CAD drawings, blueprints, videos, etc.).

CONCLUSION

Our approach has been to draw on the social and cognitive research literature in writing and upon our experience with prototype tools to identify social, cognitive and practical issues that we are attempting to address with a formative-evaluation-based prototype.

If we believe that our tool allows writers to create new forms of interaction, we need to understand the possibilities better. What new kinds of coordination structures will emerge? Are these new structures desirable? What is necessary for them to work well? We are conducting studies that chart how the prototype is used, as work-teams make progress through realistic document co-authoring and commenting tasks. This should help us come up with better descriptive theories that go beyond the normative theory that currently prevails.

McGrath (1990, p. 54) has noted how an increase in the volume of information is related to a decrease in the ability to control and structure it. Similarly, collaborative technologies that increasingly relax the boundaries of who, where, when, and what information will flow across a group network is bound to increase uncertainty. We see our work as split between increasing the technological potential of group interaction and harnessing this potential to satisfactory communication outcomes. Down this second branch, we expect to find some technological solutions, but many social ones as well. Cultures define hundreds of regulatory devices in face-to-face interaction to monitor social behavior. We are still in the earliest stages of establishing cultures for group exchange over networks.

Acknowledgments

The work reported here has been supported by NSF under grant number IRI-8902891. We thank Dale Miller and Paul Erion for work on programming the PREP editor prototype and Todd Cavalier for work on graphic design for the PREP editor interface.

REFERENCES

- ACM Special Issue on Information Filtering. (1992). *Communications of the ACM*, 35 (12), 27-81.
- Barghouti, N. S., & Kaiser, G. E. (1991). Concurrency control in advanced database applications. *Computing Surveys*, 23 (3), 269-317.
- Blakeslee, A. (1992). Inventing scientific discourse : dimensions of rhetorical knowledge in physics. Ph.D. Thesis. Carnegie Mellon University.
- Bridwell-Bowles, L. S., Johnson, P., & Brehe, S. (1987). Computers and composing: Case studies of experienced writers. In A. Matsushashi (Ed.), *Writing in real time: Modeling production processes* (pp. 81-107). Norwood, NJ: Ablex.
- Catlin, T., Bush, P., & Yankelovich, N. (1989) InterNote: Extending a hypermedia framework to support annotative collaboration. In *Hypertext'89 Proceedings* (pp. 365-378). Nov. 5-8, 1989, Pittsburgh, PA.
- Cavalier, T., Chandhok, R., Morris, J., Kaufer, D., & Neuwirth, C. M. (1991). A visual design for collaborative work: Columns for commenting and annotation. J. F. Nunamaker, Jr., (Ed.). *Proceedings of the Twenty-fourth Hawaii International Conference on System Sciences (HICSS-24)* (pp. 729-738). IEEE Press.

- Cross, G. A. (1992). A Bakhtinian exploration of factors affecting the collaborative writing of an executive letter of an annual report. *Research in the Teaching of English* 24, 2 (May, 1990), 173-203.
- Dewan, P., & Choudhary, R. (1991). Flexible user interface coupling in a collaborative system. In *Proceedings of the CHI'91 Conference* (pp. 41-49). New Orleans, April 27-May 2, 1991: ACM SIGCHI.
- Demers, A., et al. (1989). *Epidemic algorithms for replicated database maintenance*. Xerox Technical Report CSL-89-1, January.
- Dourish, P. & Bellotti, V. (1992). Awareness and coordination in shared workspaces. In *Proceedings CSCW '92 Conference on Computer-Supported Cooperative Work* (pp. 107-114). ACM SIGCHI & SIGOIS, Toronto, Oct. 31-Nov. 4, 1992.
- Edwards, M. R., Levine, J. A., & Kurland, D. M. (1986). *ForComment*. Broderbund.
- Fish, R. S., Kraut, R. E., Leland, M. D. P., & Cohen, M. (1988). Quilt: A collaborative tool for cooperative writing. In *Proceedings of COIS '88 Conference on Office Information Systems* (pp. 30-37). ACM SIGOIS.
- Flower, L., & Hayes, J.R. (1981a). A cognitive process of theory of writing. *College Composition and Communication*, 32, 365-387.
- Flower, L., & Hayes, J. R. (1981b). The pregnant pause: An inquiry into the nature of planning. *Research in the Teaching of English*, 15, 229-243.
- Flower, L., Schriver, K. A., Carey, L., Haas, C., & Hayes, J. R. (1989). *Planning in writing: The cognition of a constructive process*. Technical Report 34, Center for the Study of Writing, Carnegie Mellon University, July.
- Gabarro, J. J. (1987). The development of working relationships. In J. W. Lorsch (Ed.), *Handbook of organizational behavior* (pp. 172-189). Englewood Cliffs, NJ: Prentice Hall.
- Galegher, J. & Kraut, R. E. (1994). Computer-mediated communication for intellectual teamwork: An experiment in group writing. *Information Systems Research* 5(2), 110-138.
- Gould, J. D. How to design usable systems. In M. Helander (Ed.), *Handbook of human-computer interaction*. Elsevier Science Publishers B. V., North-Holland, Amsterdam, 1988.
- Haake, J., & Wilson, B. (1992). Supporting collaborative writing of hyperdocuments in SEPIA. In *Proceedings of CSCW'92 Conference on Computer-Supported Cooperative Work* (pp. 138-146), Oct. 31-Nov. 4, Toronto, Canada.
- Haas, C. (1989a). Seeing it on the screen isn't really seeing it: Computer writers' reading problems. In G. E. Hawisher & C. L. Selfe (Eds.), *Critical Perspectives on Computers and Composition* (pp. 16-29). New York: Teachers College Press.
- Haas, C. (1989b). How the Writing Medium Shapes the Writing Process: Effects of Word Processing on Planning. *Research in the Teaching of English* 23 (2), 181-207.
- Haas, C. (1990). Composing in technological contexts: A study of not-making. *Written Communication* 7 (4), 512-547.

- Haas, C., & Hayes, J.R. (1986). What did I just say? Reading problems in writing with the machine. *Research in the Teaching of English* 20, 22-35.
- Halasz, F. G. (1987). Reflections on NoteCards: Seven issues for the next generation of hypermedia systems. In *Hypertext'87 Proceedings* (pp. 345-365). Chapel Hill, NC, November 13-15, 1987.
- Halasz, F. G., & Schwartz, M. (1990). The Dexter hypertext reference model. In *Proceedings of the Hypertext Standardization Workshop January 16-18, 199, National Institute of Standards and Technology* (pp. 95-133). NIST Special Publication 500-178. Washington, DC: U.S. Government Printing Office.
- Hayes, J. R., & Flower, L. (1980). Identifying the organization of writing processes. In L. Gregg & E. Steinberg (Eds.), *Cognitive processes in writing: An interdisciplinary approach*. Hillsdale, NJ: Lawrence Erlbaum.
- Hayes, J. R., Flower, L., Schriver, K. A., Stratman, J., & Carey, L. Cognitive processes in revision (1987). In S. Rosenberg (Ed.), *Advances in applied psycholinguistics, Volume II: Reading, writing, and language processing*. Cambridge: Cambridge University Press.
- Hewitt, C. E. (1986). Offices are open systems. *ACM Transactions on Office Information Systems* 4 (3), 271-281.
- Kaufert, D. S., Hayes, J. R. & Flower, L. (1986). Composing written sentences. *Research in the Teaching of English* 20(2), 121-140.
- Kaufert, D. K., & Carley, C. (1993). *Communication at a distance: The influence of print on sociocultural organization and change*. Hillsdale, NJ: LEA.
- Kaye, T. (1993). Computer networking for development of distance education courses. In M. Sharples (Ed.), *Computer supported collaborative writing* (pp. 41-69). London: Springer-Verlag.
- Kistler, J. J. & Satyanarayanan, M. (1991). *Disconnected operation in the Coda file system*. Carnegie Mellon Computer Science Technical Report CMU-CS-91-166. To appear in *ACM Transactions on Computer Systems*, February, 1992.
- Kraut, R. E., Galegher, J., & Egidio, C. Relationships and tasks in scientific research collaboration. *Human-Computer Interaction* 3:31-58, 1987.
- Leland, M. D. P., Fish, R. S., & Kraut, R. E. Collaborative document production using Quilt. In *Proceedings of CSCW '88 Conference on Computer-supported Cooperative Work*, pp. 206-215. ACM SIGCHI & SIGOIS, Portland, OR, September 26-28, 1988.
- Malone, T. W. What is coordination theory? In *Coordination Theory Workshop*. National Science Foundation, Feb., 1988.
- Malone, T. W., & Crowston, K. (1994). The interdisciplinary theory of coordination. *ACM Computing Surveys*, 26 (1), 87-199.
- McGrath, J. E. (1990). Time matters in groups. In J. Galegher, R. E. Kraut, & C. Egidio (Eds.), *Intellectual teamwork* (pp. 23-61). Hillsdale, NJ: Lawrence Erlbaum.

McLaughlin Hymes, C., & Olson, G. M. (1992). Unblocking brainstorming through the use of a simple group editor. In *Proceedings CSCW '92 Conference on Computer-Supported Cooperative Work* (pp. 99-106). ACM SIGCHI & SIGOIS, Toronto, Oct. 31-Nov. 4, 1992.

Minör, S., & Magnusson, B. (1993). A model for semi-(a)synchronous collaborative editing. In *Proceedings of the 3rd European Conference on Computer-Supported Cooperative Work* (pp. 219-231), Sep 13-17, 1993, Milan, Italy.

Neuwirth, C.M., Chandhok, R., Charney, D., Wojahn, P., Kim, L. (1994). Distributed collaborative writing: A comparison of spoken and written modalities for reviewing and revising documents. In *Proceedings of the CHI'94 Conference on Computer-Human Interaction* (pp. 51-57), April 24-28, Boston. ACM.

Neuwirth, C. M., Kaufer, D. S., Chimera, R., & Gillespie, T. (1987). The Notes program: A hypertext application for writing from source texts. In *Hypertext'87 Proceedings* (pp. 345-365). Chapel Hill, NC, November 13-15, 1987.

Neuwirth, C. M., Kaufer, D. S., Keim, G., & Gillespie, T. (1988). *The Comments program: Computer support for response to writing*. CECE-TR-3, Center for Educational Computing in English, English Department, Carnegie Mellon University, January, 1988.

Neuwirth, C. M., & Kaufer, D. S. (1989). The role of external representations in the writing process: Implications for the design of hypertext-based writing tools. In *Hypertext '89 Proceedings* (pp. 319-342). Pittsburgh, PA, November 5-8, 1989.

Neuwirth, C. M., & Kaufer, D. S. (1992). Computers and composition studies: Articulating a pattern of discovery. In G. E. Hawisher and P. LeBlanc (Eds.) *Reimagining computers and composition: Teaching and research in the virtual age* (pp. 173-190). Portsmouth, NH: Boynton/Cook Publishers, Inc.

Neuwirth, C. M., Kaufer, D. S., Chandhok, R., & Morris, J. H. (1990). Issues in the design of computer-support for co-authoring and commenting. *Proceedings of the Third Conference on Computer-Supported Cooperative Work (CSCW '90)* (pp. 183-195). Baltimore, MD: Association for Computing Machinery.

Neuwirth, C. M., Palmquist, M., & Gillespie, T. Role playing in peer review: The Devil's Advocate exercise. In D. Beil (Ed.), *Teacher's guide to using computer networks for written instruction*, pp. 157-164. Realtime Learning Systems, Washington, DC, 1989.

Newman, R., & Newman, J. (1993). Social writing: Premises and practices in computerized contexts. In M. Sharples (Ed.), *Computer supported collaborative writing* (pp. 29-40). London: Springer-Verlag.

Norman, D. A. (1986). Cognitive engineering. In D. A. Norman & S. W. Draper (Eds.), *User-centered system design* (pp. 31-61). Hillsdale, NJ: Lawrence Erlbaum Associates.

Olson, G. M., & Olson, J.S. (1991). User-centered design of collaboration technology. *Journal of Organizational Computing*, 1, 61-83.

Olson, J. S., Olson, G. M., Storrøsten, M., & Carter, M. (1992). How a group-editor changes the character of a design meeting as well as its outcome. In *Proceedings CSCW '92 Conference on Computer-Supported Cooperative Work* (pp. 91-98). ACM SIGCHI & SIGOIS, Toronto, Oct. 31-Nov. 4, 1992.

Posner, I.R., & Baecker, R.M. (1992). How people write together. In *Proceedings of the 25th Hawaii International Conference on System Sciences*, Vol. 4, Hawaii, 1992.

Smith, J. B., Weiss, S. F., & Ferguson, G. J. (1987). A hypertext writing environment and its cognitive basis. In *Hypertext'87 Proceedings* (pp. 345-365). Chapel Hill, November 13-15, 1987.

Smolensky, P., Bell, B., Fox, B., King, R., & Lewis, C. (1987). Constraint-based hypertext for argumentation. In *Hypertext'87 Proceedings* (pp. 215-246). Chapel Hill, November 13-15, 1987.

Tang, J. C., & Leifer, L. J. (1988). A framework for understanding the workspace activity of design teams. In *Proceedings CSCW '88 Conference on Computer-Supported Cooperative Work* (pp. 244-249). ACM SIGCHI & SIGOIS, Portland, September 26-29, 1988.