A MINIMALIST APPROACH TO THE DEVELOPMENT OF A WORD PROCESSOR SUPPORTING GROUP WRITING ACTIVITIES

Nicholas Malcolm and Brian R. Gaines
Knowledge Science Institute, University of Calgary
Calgary, Alberta, Canada T2N 1N4.
malcolm@cs.tamu.edu, gaines@cpsc.ucalgary.ca

Group-writing, in which a document is jointly produced by a team of writers, occurs widely in science and in industry. There are now a number of products and research tools designed to support group-writing teams. Most, however, require use of non-mainstream word-processing systems, and assume that full information is available through a network to mediate conflicts. This paper reports research on group-writing tools that deviate as little as possible from conventional word processors and assume only intermittent network connection for document exchange and conflict resolution. GroupWriter can be used by some people in a collaborative team as a conventional word processor, by others as a versioning and text/sound annotation system, and by others as a full hypertext system all while working with the same corpus of documents. It offers full typographic and page layout facilities and imports typographic text from, and exports to, the mainstream commercial word processors.

1 INTRODUCTION

Science is a collaborative process, with researchers building upon results obtained by their forerunners. In particular, an increasing number of research papers are being jointly authored (Over, 1982). Group-writing systems will be part of the NSF's proposed National Collaboratory (Lederberg & Uncapher, 1989)—a high speed network with facilities designed to support collaboration between scientists. In the business community there is also a large demand for group-writing tools (Dalton, 1987; Stratton, 1989). Many organizations need to manage the evolution of documents from draft form to final versions, often passing through the hands of many people along the way.

While the joint production of documents may be seen as a major organizational function, the availability and acceptance of systems to support it lags way behind the current capabilities of network and workstation technology. We hypothesized that the main potential users of GroupWriter would be current users of standard commercial word processors. They would not be ready to accept any degradation in facilities in using an experimental system and neither would they be willing to make major changes in their work practice—certainly not in the short term. Thus a primary objective of system development in this area must be the minimalist extension of existing word processing practice to incorporate effective support for group writing.

This paper shows that such a minimalist approach is feasible and results in word processing technology that offers both the full functionality expected in a word processor, and that necessary for version and annotation control in group writing, while deviating very little from current word processing practice.

2 GROUP WRITING SYSTEMS

Several group-writing systems have been implemented, including Contexts (Delisle & Schwartz, 1987), Shared Books (Lewis & Hodges, 1988), CES (Seliger, 1985), and PREP (Neuwirth, Kaufer, Chandhok & Morris, 1990). The widely used Revision Control System (RCS) (Tichy, 1982), which manages multiple revisions of text files may be seen as a basic group-writing system. Quilt (Fish, Kraut & Leland, 1988; Leland, Fish & Kraut, 1988) is a computer-based tool for collaborative document production, and provides structured facilities for either voice or text annotation of documents. A history version of the document, complete with annotations, can be saved at any time. Using automatic paragraph comparison, readers can use history versions to see side by side comparisons of alternative versions of a paragraph. If two or more users simultaneously attempt to edit the same section of a document, a warning is given. If ignored, alternative versions must be merged manually.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.
© 1991 ACM 0-89791-456-2/91/0010/0147...$1.50

Malcolm, N. and Gaines, B. R. (1991). A minimalist approach to the development of a word processor supporting group writing activities. In COCS'91: Proceedings of Conference on Organizational Computing Systems, pp. 147-152. ACM Press.

Several hypertext systems providing support for collaborative writing have also been implemented. Contexts (Delisle & Schwartz, 1987) is an extension of the Neptune hypertext environment, allowing users their own private views or contexts of the hypertext database. ForComment (Dalton, 1987) allows authors to attach annotations to any of the lines of text in a document. InterNote (Catlin, Bush & Yankelovich, 1989) extends the Intermedia environment to provide better support for annotation. It tracks the author of each annotation, and allows the contents of an annotation to be easily incorporated into the text of the main document. However neither ForComment nor InterNote support document versioning, in which multiple versions of a document may coexist. GroupWriter is designed to manage versions at both the document and the paragraph level.

Contexts, Quilt, and PREP all support document versioning in addition to annotation. Quilt attempts to manage the complexities of collaborative writing by providing structured annotation facilities and social roles for authors. PREP allows free-floating hypertext nodes during the planning phase of a writing project. The nodes can then be arranged in a grid, one column for the document plan, another for the document text, and the remainder for annotations. Contexts and Quilt, however, are like traditional hypertext systems in which hypertext links disrupt the flow of the document text. PREP avoids this problem by laying out the document and the annotations in a grid. However, PREP has limited formatting capabilities, and hence writers cannot see their document in final form. GroupWriter uses a high-quality, commercially available word-processing engine to provide typographic quality formatting and output, and supplements this with hypertext annotation in the margins in such a way as to not disturb the integrity of the text.

RCS runs on Unix networks and Quilt under X-windows, and use the editors available in the Unix environment. GroupWriter is designed to run on personal computers intermittently connected to a network and to be interface-compatible with Microsoft Word, and data-compatible with word processors having RTF import/export.

3 GROUPWRITER REQUIREMENTS SPECIFICATION

In designing GroupWriter, particular attention was paid both to cognitive aspects of writing and to the way in which collaborative writing groups are structured. During the past decade, cognitive models of writing have generated much interest (Nystrand, 1986; Sharples & Pemberton, 1988). Most cognitive models view writing as an under-constrained goal oriented activity, and highlight the central role of writer intention, purpose, plans, etc. in the composing process. In addition, efforts have been made to understand the impact of computers on the writing process, and to modify cognitive models accordingly (Sharples & Pemberton, 1990).

From the previous systems and literature on collaborative writing it is possible to determine the significant requirements to be satisfied by a group-writing system which go beyond those of a single-user editing system:

1. *Reproduction of older versions.* It must be possible to trace the evolution of a document, and to revert to a previous version if subsequent changes are deemed inadequate.
2. *Maintaining multiple parallel versions of a document.* The existence of multiple parallel versions of a document must be allowed. For example, several users may be simultaneously editing the document, or there may be two versions of a scientific paper, one for a conference and one for a journal.
3. *Comparison of versions.* The option of comparing alternative or previous versions of a document should be supplied, highlighting differences between the text of different versions (Dalton, 1987).
4. *Record comments and suggested revisions.* Comments and suggested revisions of portions of text made by members of a group-writing team should be recorded and easily accessible (Dalton, 1987). Additionally, these annotations should not disrupt the text of the original document.
5. *Compatibility with other editors.* In order for a group-writing system to be widely used, it should be compatible with popular existing editors. In a survey of fifty pairs of collaborative researchers, Kraut, Galegher and Egido (1987) found that most had problems with incompatibilities among programs and computing environments. This often lead to one partner controlling the document and incorporating the other's handwritten changes into the draft.

6. *Simple and natural to use.* The majority of potential users of a group-writing system have a nontechnical background and are already familiar with commercial word processors. An acceptable group-writing system has to be very simple and natural to use, and expectations created by single-user systems such as Microsoft Word need to be supported.
7. *No requirement to manage complexity.* Users should not be required to understand and manage the versioning system. Some users may wish to edit documents creating revisions which others in the group manage as alternative versions. For such users the system should appear to be a conventional word processor.
8. *Support of electronic mail.* Electronic mail facilities should be integrated with a group-writing system to facilitate communication between writers.
9. *Reliably store information.* Changes to a document must be reliably stored despite computer or network failure (Lewis & Hodges, 1988).
10. *Protection of privacy.* Access by unauthorized people must be preventable, because loss or corruption of material may have financial or legal consequences (Lewis & Hodges, 1988).

4 GROUPWRITER DOCUMENT REPRESENTATION

The structure of a document version is maintained at the paragraph level. The paragraph was chosen as the unit of granularity because it is both a natural conceptual unit for document producers (Halliday & Hasan, 1976) and a basic data structure in word processors. Annotations are stored in the same way as paragraphs. Each paragraph has a unique identification (ID) number, and a link to the paragraph from which it derived. This results in a paragraph derivation tree as shown in Figure 1.

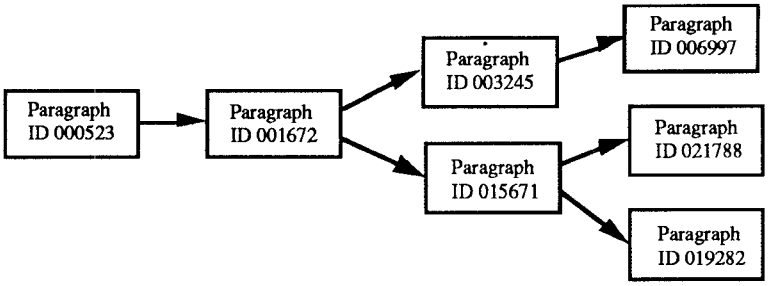

Figure 1 Paragraph Derivation Tree

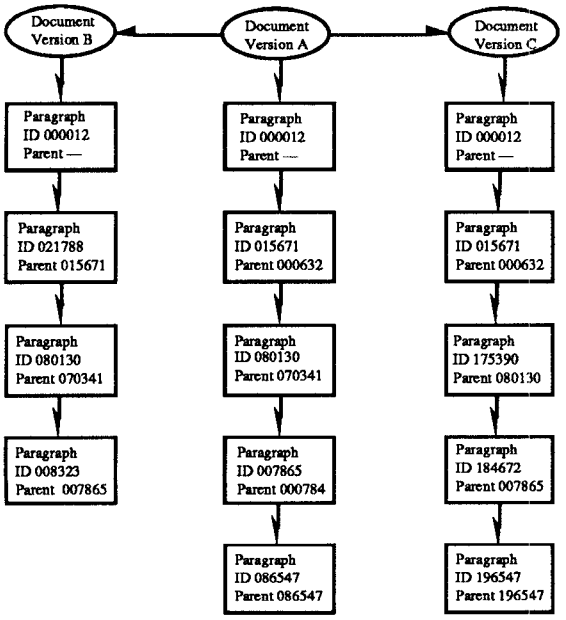

Figure 2 Document Version Structure

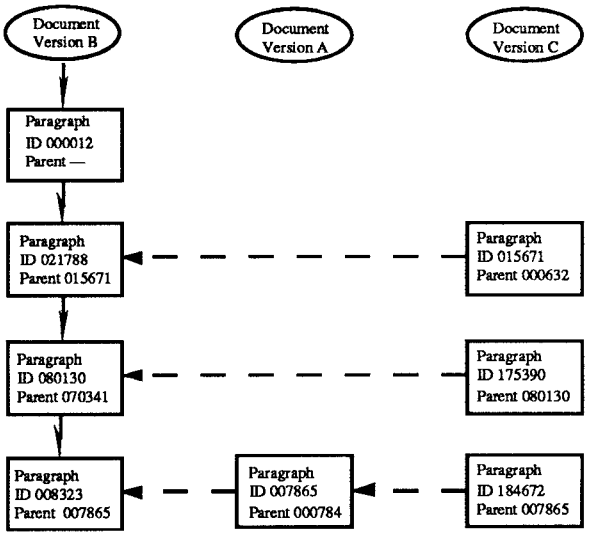

Figure 3 Alternative Paragraphs
(B is the primary version, and C the secondary—paragraphs in C with common derivations to those in B are available as alternatives)

A document version is identified by name, and its structure is stored as a linked list of paragraphs together with a link to the document version from which it derived (Figure 2). At the beginning of an editing or merging session a user selects a document version, designated the primary version. Additionally, other versions of a document may be designated secondary versions. Secondary version paragraphs that are alternatives of paragraphs in the primary version (Figure 3) are available during editing. They may be modified and incorporated into the primary document version. During editing each paragraph has an associated flag to indicate whether it has been modified.

5 USER INTERFACE DESIGN

Given the inherent complexity of hypertext, and of collaborative writing itself, the user-interface to a hypertext collaborative writing system is extremely important. The critical features that must be clearly presented to the user are the existence of annotations and the relations between different versions of the document. In addition, GroupWriter is designed in such a way that the same document may be viewed in different ways depending on the expertise and the inclination of the user. It may be used simply as a conventional word processor, as a word processor with versioning and annotation facilities, or as a full hypertext system. This allows a group to structure itself such that some writers can create and edit document versions conventionally while others take responsibility for reconciling versions.

Support for managing document versions is structured such that users can use facilities similar to those in conventional word processors. They can track versions through naming conventions if they wish and open both primary and secondary versions through a standard file opening dialog. However, a graphic browser showing relationships between document versions is also provided (Figure 4). Double-clicking on a document version opens it for editing as the primary document version, with any other selected versions being opened as secondary versions.

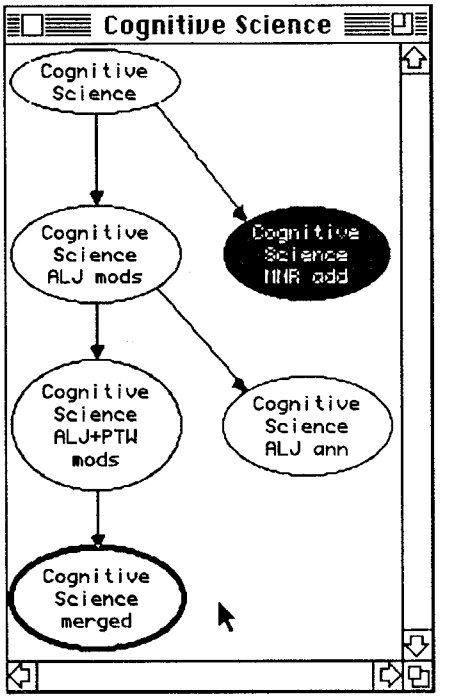

Fig. 4 Document Version Browser

The primary document version appears in a normal edit window as shown in Figure 6 with lines in the left margin indicating the existence of alternative paragraphs or annotations. The mouse pointer changes shape when moved over the line, to indicate that a popup menu is available. The menu provides a number of operations that may be performed on annotations and alternative paragraphs. They may be displayed in a separate window by the **Show** operation; inserted into the document text by the **Place Before**, **Place After**, or the **Replace** operations; removed by the **Remove** operation. Sound annotations may be heard by selecting the **Play** operation.

When **Merge** is selected alternative paragraphs are compared on a sentence-by-sentence basis with the corresponding paragraph in the document version being edited, and the differences displayed in a separate window (Figure 5). Differing sentences are marked by brackets in the left margin, the thick bracket indicating that the sentence originated in the document version being edited. Sentences to be included in the merged paragraph are selected by clicking in the corresponding bracket. The merge is completed by double-clicking in one of the brackets, at which time the merged paragraph is inserted into the document version being edited.

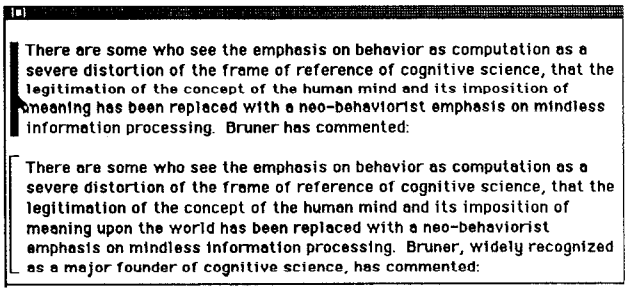

Figure 5 Merging two Paragraphs

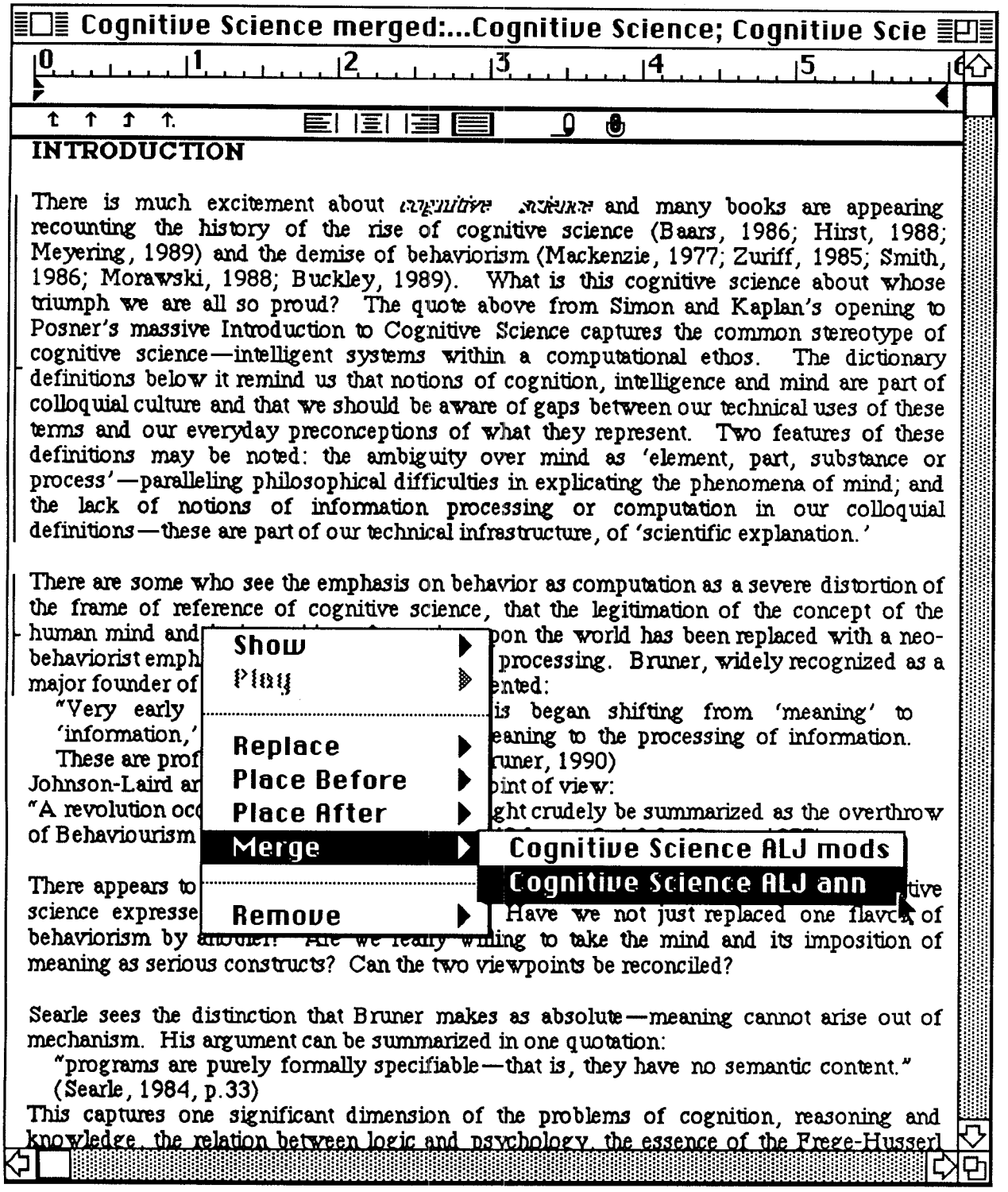

Figure 6 Alternative Paragraph Markers and the Popup Menu

Paragraphs may have either text or sound annotations, and are annotated by placing the cursor within the paragraph to be annotated and either selecting the appropriate menu item or clicking on the appropriate icon in the ruler.

6 CONCLUSIONS

The development of GroupWriter is part of a program of research on Knowledge Support Systems (Gaines, 1990) to support organizations at a wide range of levels from electronic communication, through document production, to hypermedia, knowledge modeling and reasoning.

Current networking, workstation and software engineering technologies make it feasible to develop word processors supporting collaborative writing processes within organizations. It is feasible to offer the full functionality required with minimal extensions to current word processing user interfaces requiring little change in work practice. This is an effective way of supporting a versioning and annotation system for the production of documents by a group of people.

ACKNOWLEDGEMENTS

Financial assistance for this work has been made available by the Natural Sciences and Engineering Research Council of Canada. We are grateful to Saul Greenberg and Mildred Shaw for critical comments that have improved the GroupWriter interface, and to colleagues who have participated in user trials for GroupWriter.

REFERENCES

Catlin, T., Bush, P. & Yankelovich, N. (1989). InterNote: Extending a hypermedia framework to support annotative collaboration. **Hypertext'89 Proceedings**, pp. 365-378. Pittsburgh, Pennsylvania (November).

Dalton, R. (1987). Group-writing tools: Four that connect. **Information Week**, pp. 62-65, (March).

Delisle, N.M. & Schwartz, M.D. (1987). Contexts—A partitioning concept for hypertext. **ACM Transactions on Office Information Systems, 5**(2), pp. 168-186 (April).

Fish, R.S., Kraut, R.E. & Leland, M.D.P. (1988). Quilt: A collaborative tool for cooperative writing. **Proceedings of the Conference on Office Information Systems**. pp. 30-37. Palo Alto, California (March).

Gaines, B.R. (1990). Knowledge support systems. **Knowledge-Based Systems 3**(4) 192-203 (December).

Halliday, M.A.K. & Hasan, R. (1976). **Cohesion in English**. London: Longmans.

Kraut, R.E, Galegher, J. & Egido, C. (1987). Relationships and tasks in scientific research collaboration. **Human-Computer Interaction. 3**, pp. 31-58, 1987.

Lederberg, J. & Uncapher, K. (1989). Towards a National Collaboratory. Report of an Invitational Workshop at The Rockefeller University (March).

Leland, M.D.P., Fish, R.S. & Kraut, R.E. (1988). Collaborative document preparation using Quilt. **Proceedings of the Conference on Computer-Supported Cooperative Work**, pp. 206-215. Portland, Oregon (September).

Lewis, B.T. & Hodges, J.D. (1988). Shared Books: Collaborative publication management for an office information system. **Proceedings of the Conference on Office Information Systems,** pp. 197-204, Palo Alto, California (March).

Neuwirth, C.M., Kaufer, D.S., Chandhok, R. & Morris, J.J. (1990). Issues in the design of computer support for co-authoring and commenting. **Proceedings of the Third Conference on Computer Supported Cooperative Work**, pp. 183-195, Los Angeles, California (October).

Nystrand, M. (1986). **The Structure of Written Communication: Studies in Reciprocity between Writers and Readers.** Academic Press, 1986.

Over, R. (1982). Collaborative research and publication in psychology. **American Psychologist, 37**(9), pp. 996-1001 (September).

Seliger, R. (1985). Design and implementation of a distributed program for collaborative editing. Master's thesis, Massachusetts Institute of Technology (September).

Sharples, M. & Pemberton, L. (1988). Representing writing: An account of the writing process with regard to the writer's external representations. Cognitive Science Research Paper CSRP 119, School of Cognitive Sciences, The University of Sussex.

Sharples, M. & Pemberton, L. (1990). Starting from the writer guidelines for the design of user-centred document processors. Cognitive Science Research Paper CSRP 154, School of Cognitive and Computing Sciences, The University of Sussex.

Stratton, C.R. (1989). Collaborative writing in the workplace. **IEEE Transactions on Professional Communication**, 32(3), pp. 178-182 (September).

Tichy, W.F. (1982). Design, implementation and evaluation of a revision control system. **Proceedings of the 6th International Conference on Software Engineering** (September).